Documentation

 **Requirements Documentation:**

* **Functional Requirements**: Describe the functionalities the application must fulfill.
* **Non-functional Requirements**: Cover aspects like performance, security, and usability.
* **Technical Requirements**: Specify the software, hardware, and network requirements.

 **Design Documentation:**

* **Architecture Design**: Provide an overview of the application's architecture, including how the Windows Forms front end interacts with the SQL Server back end.
* **Database Design**: Include Entity-Relationship Diagrams (ERDs), database schema, and a description of the tables, relationships, indexes, and stored procedures.
* **Class Diagrams**: Illustrate the object-oriented design of the application, showing classes, their properties, methods, and relationships.
* **UI/UX Design**: Wireframes or mockups of the user interface, along with design decisions and user flow.

 **Source Code Documentation:**

* **Inline Comments**: Document the code with inline comments to explain complex logic, classes, methods, and key decision points.
* **Code Documentation**: Use XML documentation comments for methods, classes, and properties, providing summaries, parameter descriptions, and return values.

 **Configuration Documentation:**

* **Database Connection Strings**: Document the configuration for connecting to SQL Server, including security settings.
* **App Configurations**: Document any configuration settings in app.config or web.config files.

 **Testing Documentation:**

* **Test Cases/Plans**: List the test cases that need to be executed, including functional tests, integration tests, and performance tests.
* **Bug Tracking**: Maintain a log of bugs and issues found during testing, along with their status and resolution.

 **User Documentation:**

* **User Manual**: Instructions on how to install, configure, and use the application.
* **Troubleshooting Guide**: Common issues users may encounter and steps to resolve them.

 **Deployment Documentation:**

* **Deployment Guide**: Instructions for deploying the application, including prerequisites, installation steps, and configuration.
* **Backup and Recovery Procedures**: Steps for backing up the database and recovering it in case of failure.

 **Maintenance Documentation:**

* **Change Log**: Record of changes made to the application over time, including new features, bug fixes, and updates.
* **Support Documentation**: Information for technical support, including how to report issues and where to find help.

**Functional Requirements for CEA Application**

**1. User Authentication and Authorization**

* **1.1. User Sign-In:**
  + **1.1.1.** The application must provide a sign-in screen where the user can enter a username and password.
  + **1.1.2.** The system must validate the user's credentials against the stored data in the SQL Server database.
  + **1.1.3.** If the credentials are correct, the user is granted access to the application. If the credentials are incorrect, an error message must be displayed, and the user should be allowed to retry.
  + **1.1.4.** The system must enforce password security policies (e.g., minimum length, and complexity requirements).
  + **1.1.5.** The application should support account lockout after a predefined number of failed sign-in attempts.
* **1.2. Role-Based Access Control:**
  + **1.2.1.** Upon successful sign-in, the system must retrieve the user's role and permissions from the database.
  + **1.2.2.** The system must restrict access to specific screens and functionalities based on the user's role.
  + **1.2.3.** Users with administrative roles must have access to a user management screen where they can create, modify, or delete user accounts, assign roles, and manage permissions.
  + **1.2.4.** The system must log all sign-in attempts, successful or unsuccessful, for audit purposes.

**2. Purchase Order Request Creation**

* **2.1. Open Purchase Order Request Screen:**
  + **2.1.1.** After successful sign-in, the user must be presented with a dashboard containing a menu or navigation pane to access the Purchase Order Request screen.
  + **2.1.2.** The Purchase Order Request screen must be accessible only to users with the appropriate permissions (e.g., purchasing employees).
  + **2.1.3.** The screen must provide options to create a new purchase order request or view/edit existing requests.
* **2.2. Input Mandatory Fields:**
  + **2.2.1.** The system must display a form where the user can input the following mandatory fields for the purchase order:
    - **2.2.1.1.** Number of Request: A unique identifier for the purchase request.
    - **2.2.1.2.** Type of Purchase: A dropdown or selection list specifying the type of purchase (e.g., Goods, Services, etc.).
    - **2.2.1.3.** Number of Contract: The contract number associated with the purchase.
    - **2.2.1.4.** Purchase Value: The total monetary value of the purchase, which must be greater than 1 million Mexican pesos.
    - **2.2.1.5.** Company Name: The name of the company making the purchase.
    - **2.2.1.6.** Requester Name: The name of the individual or department requesting the purchase.
    - **2.2.1.7.** Direction of Request: The department or division within the company that is making the request.
    - **2.2.1.8.** Title of the Purchase: A brief title summarizing the nature of the purchase.
    - **2.2.1.9.** Description of the Purchase: A detailed description of the purchase, including any specific requirements or conditions.
  + **2.2.2.** All mandatory fields must be validated to ensure they are not left blank. The system must prompt the user to complete any missing fields before allowing the submission of the request.
  + **2.2.3.** The system must provide input validation for each field (e.g., numeric fields should only accept numbers, text fields should have a character limit, etc.).
  + **2.2.4.** The system must save the inputted data to the SQL Server database, associating it with the user's account and a timestamp.
* **2.3. Upload Image of Comparison Table:**
  + **2.3.1.** The Purchase Order Request screen must include an option to upload an image file containing a comparison table between vendors and their respective prices.
  + **2.3.2.** The system must accept common image formats (e.g., JPG, PNG) and validate that the uploaded file meets size and format requirements.
  + **2.3.3.** The user must be able to preview the uploaded image before submitting the purchase order request.
  + **2.3.4.** The system must store the uploaded image in the database or a secure file storage system, linking it to the corresponding purchase order request.
  + **2.3.5.** The system must allow the user to replace or delete the uploaded image before final submission if needed.
  + **2.3.6.** The system must generate a unique identifier for each uploaded image and associate it with the purchase order record.
* **2.4. Submit Purchase Order Request:**
  + **2.4.1.** Once all mandatory fields are completed, and the image is uploaded, the user must have the option to submit the purchase order request.
  + **2.4.2.** Upon submission, the system must validate all data and provide a summary of the entered information for the user to review.
  + **2.4.3.** The system must generate a confirmation message upon successful submission, including the purchase order number and a timestamp.
  + **2.4.4.** The system must log the submission details for audit and tracking purposes.

**3. Purchase Order Request Review and Approval Process**

* **3.1. Review Screen for Authorized Users:**
  + **3.1.1.** The system must provide a separate screen accessible only to authorized users, such as supervisors and other personnel involved in the approval process.
  + **3.1.2.** The screen must display a list of all purchase order requests pending review. Each entry in the list must include key fields from the mandatory fields, such as:
    - Number of Request
    - Type of Purchase
    - Purchase Value
    - Company Name
    - Title of the Purchase
    - Current Approval Status (e.g., Pending Department Approval, Pending Management Approval, etc.)
  + **3.1.3.** The list must be sortable and filterable based on different criteria (e.g., by department, by approval status, by date).
  + **3.1.4.** Authorized users must be able to select a purchase order from the list to view detailed information and proceed with the review.
* **3.2. Detailed Purchase Order View:**
  + **3.2.1.** Upon selecting a purchase order from the list, the system must display a detailed view of the request, including all the information entered during the purchase order request creation process.
  + **3.2.2.** The detailed view must include the uploaded image of the comparison table between vendors and prices.
  + **3.2.3.** The layout of this screen should mirror the design of the Purchase Order Request creation screen to maintain consistency.
  + **3.2.4.** The system must display the current approval level and history of actions taken (e.g., who approved, who modified, or who rejected the request, including timestamps).
* **3.3. Four-Step Approval Process:**
  + **3.3.1.** The purchase order request must go through a four-step approval process:
    - **3.3.1.1.** **Department Approval:** The department responsible for the purchase must review and approve the request.
    - **3.3.1.2.** **Management Approval:** Once approved by the department, the request moves to management for review and approval.
    - **3.3.1.3.** **Direction Approval:** After management approval, the request is forwarded to the direction level for approval.
    - **3.3.1.4.** **General Direction Approval:** The final approval is made by the general direction of the company.
  + **3.3.2.** Each approval step must be handled by the relevant authorized user, who must have the following options:
    - **3.3.2.1.** **Approve:** The system must move the request to the next approval level and record the approval action in the approval history.
    - **3.3.2.2.** **Modify:** The user must be able to modify specific details of the request if needed. Any modifications must be recorded in the approval history, and the request must stay at the current approval level for further review.
    - **3.3.2.3.** **Reject:** If the request is rejected, the system must move the request back to the previous approval level. The rejection must be recorded in the approval history, including the reason for rejection.
* **3.4. Tracking and Notifications:**
  + **3.4.1.** The system must track the status of each purchase order request throughout the approval process, updating the status in real-time.
  + **3.4.2.** The system must send notifications to the relevant users when an action is required (e.g., when a request is ready for their approval).
  + **3.4.3.** The system must log all actions taken during the review process, including who took the action, what action was taken, and when it occurred.
* **3.5. Final Submission to General Direction:**
  + **3.5.1.** Once the request receives all four approvals, the system must mark the request as "Approved for General Direction Review."
  + **3.5.2.** The request must then be available to the general direction for final consideration.

**4. Search and History Module**

* **4.1. Search Functionality:**
  + **4.1.1.** The system must provide a search interface that allows users to search through the history of purchase order requests.
  + **4.1.2.** Users must be able to search by various criteria, including but not limited to:
    - Number of Request
    - Type of Purchase
    - Purchase Value
    - Company Name
    - Date Range
    - Approval Status
  + **4.1.3.** The search results must display relevant details for each request and provide an option to view the full details of any selected request.
  + **4.1.4.** The system must allow exporting search results to a file format such as CSV or Excel for further analysis.
* **4.2. History Tracking:**
  + **4.2.1.** The system must maintain a complete history of all purchase order requests, including all actions taken (e.g., approvals, modifications, rejections) with timestamps.
  + **4.2.2.** Users with appropriate permissions must be able to view the full history of any request, including the sequence of approvals and any modifications made.

**5. Reporting Module**

* **5.1. Report Generation:**
  + **5.1.1.** The system must provide a reporting module where users can generate various reports based on purchase order requests from previous sessions.
  + **5.1.2.** Reports must be customizable, allowing users to select specific fields and filters (e.g., date range, department, approval status) to generate insights.
  + **5.1.3.** The system must support different types of reports, such as summary reports, detailed reports, and trend analysis.
  + **5.1.4.** Generated reports must be viewable within the application and exportable to common file formats such as PDF, Excel, or Word.
* **5.2. Session Reporting:**
  + **5.2.1.** The system must allow users to create and generate reports for specific sessions (groups of requests).
  + **5.2.2.** These session reports must summarize the requests included in the session and show the overall status of the session (e.g., how many requests were approved, rejected, or pending).

**6. Tools Module: Session Management**

* **6.1. Create and Manage Sessions:**
  + **6.1.1.** The system must include a tools module where users can create sessions to group multiple purchase order requests.
  + **6.1.2.** Users must be able to add or remove requests from a session, providing a summary of the requests included.
  + **6.1.3.** Each session must have a unique identifier and a title, with the ability to track the overall approval status of the session.
* **6.2. Presenting Sessions to General Direction:**
  + **6.2.1.** Once a session is created, users must be able to present it to the general direction for final approval or rejection.
  + **6.2.2.** The system must provide a view where the general direction can review all requests within the session, make decisions, and provide feedback.

**7. Crystal Reports Module**

* **7.1. Report Printing:**
  + **7.1.1.** The system must integrate with Crystal Reports to generate print-ready reports summarizing purchase order requests.
  + **7.1.2.** The report must include all relevant details of the requests, including approval history, and be formatted for physical signatures by the general directors.
  + **7.1.3.** Users must be able to print these reports directly from the application or export them for printing elsewhere.

**8. Session Closure Module**

* **8.1. Closing Sessions:**
  + **8.1.1.** After the general direction has reviewed a session, the system must provide a module to close the session.
  + **8.1.2.** The system must record the final decisions for each request in the session (approved or rejected) and mark the session as closed.
  + **8.1.3.** The system must prevent further modifications to a closed session, ensuring the integrity of the final decisions.
  + **8.1.4.** Closed sessions must still be viewable in the history module for audit and reference purposes.

**Non-Functional Requirements for CEA Application**

**1. Performance Requirements**

* **1.1. Response Time:**
  + The application must respond to user actions (e.g., screen navigation, form submissions) within 2 seconds under normal load conditions.
  + Reports and searches must be generated within 5 seconds for typical queries and within 15 seconds for more complex queries involving large datasets.
* **1.2. Scalability:**
  + The system must be able to handle up to 1,000 concurrent users without degradation in performance.
  + The system must be able to manage up to 10,000 purchase order requests without performance issues.
* **1.3. Throughput:**
  + The application should be capable of processing up to 100 purchase order submissions per minute during peak usage.

**2. Reliability Requirements**

* **2.1. Availability:**
  + The system must have an availability of 99.9%, ensuring minimal downtime throughout the year.
* **2.2. Data Integrity:**
  + The system must ensure that all data entered, processed, and stored is accurate and consistent. Data integrity checks must be performed before any data is committed to the database.
* **2.3. Fault Tolerance:**
  + The application must be able to recover from hardware failures, with automatic failover to a backup system within 5 minutes.

**3. Security Requirements**

* **3.1. Authentication:**
  + The application must enforce strong password policies, including a minimum of 8 characters, a combination of uppercase, lowercase, numbers, and special characters.
  + The system must support multi-factor authentication (MFA) for all users, especially for those with higher-level access (e.g., managers, directors).
* **3.2. Authorization:**
  + Access to the system must be role-based, with permissions assigned according to user roles.
  + Sensitive operations (e.g., approvals, rejections, session closures) must require elevated permissions and must be logged for audit purposes.
* **3.3. Data Encryption:**
  + All sensitive data, including user credentials and purchase order details, must be encrypted both in transit (using SSL/TLS) and at rest.
* **3.4. Logging and Monitoring:**
  + The system must log all user activities, especially critical actions like approvals, rejections, and session closures. These logs must be stored securely and accessible only to authorized personnel.
  + The system must be monitored continuously for security breaches, with alerts sent to the appropriate personnel in case of suspicious activities.

**4. Usability Requirements**

* **4.1. User Interface:**
  + The user interface must be intuitive, with a consistent layout across all screens.
  + The application must support keyboard shortcuts for common actions to improve user efficiency.
* **4.2. User Documentation:**
  + The application must include comprehensive user manuals and help files accessible from within the application.
  + Tooltips and inline help must be provided for complex fields and operations.
* **4.3. Accessibility:**
  + The application must comply with accessibility standards, such as WCAG 2.1, to ensure that it is usable by people with disabilities.

**5. Maintainability Requirements**

* **5.1. Code Quality:**
  + The application code must be modular, well-documented, and adhere to coding standards (e.g., Microsoft .NET coding guidelines).
* **5.2. Error Handling:**
  + The system must include comprehensive error handling, with user-friendly error messages and detailed logging of technical issues.
* **5.3. Extensibility:**
  + The system must be designed in a way that allows for future enhancements without significant changes to the existing codebase.

**6. Portability Requirements**

* **6.1. Operating System Compatibility:**
  + The application must be compatible with Windows 10 and later versions.
* **6.2. Database Compatibility:**
  + The application must support Microsoft SQL Server 2016 and later versions.
* **6.3. Environment Independence:**
  + The application must be able to be deployed in both on-premises and cloud environments without requiring significant changes.

**7. Backup and Recovery Requirements**

* **7.1. Data Backup:**
  + The system must perform automated daily backups of all critical data, including user data, purchase orders, and logs.
* **7.2. Recovery Time Objective (RTO):**
  + In the event of a system failure, the system must be able to recover and be operational within 4 hours.
* **7.3. Recovery Point Objective (RPO):**
  + The system must ensure that no more than 1 hour of data is lost in the event of a system failure.

**8. Compliance Requirements**

* **8.1. Regulatory Compliance:**
  + The application must comply with relevant data protection regulations, such as GDPR for any European users, and any local regulations specific to Mexico.
* **8.2. Audit Trails:**
  + The system must maintain comprehensive audit trails of all user actions, particularly those involving financial transactions and approvals, to comply with internal and external audits.

**Technical Requirements**

**1. Development Environment**

* **1.1. Integrated Development Environment (IDE):**
  + Microsoft Visual Studio 2013 for development.
* **1.2. Programming Language:**
  + C# as the primary programming language.
* **1.3. .NET Framework:**
  + .NET Framework 4.5, which was the current version in 2013.
* **1.4. Source Control:**
  + Team Foundation Server (TFS) 2013 for version control, if available, or Git with a local setup for repository management.

**2. Database Requirements**

* **2.1. Database Management System:**
  + Microsoft SQL Server 2012 or SQL Server 2014 for database management.
* **2.2. Database Design Tools:**
  + SQL Server Management Studio (SSMS) 2012 or 2014 for managing and querying the SQL Server database.
* **2.3. Database Access:**
  + ADO.NET for database connectivity and SQL queries within the application.
* **2.4. Database Security:**
  + Basic encryption for sensitive data fields, possibly using SQL Server's built-in encryption functions.
* **2.5. Backup and Recovery:**
  + SQL Server Agent for scheduling backups, with manual failover procedures.

**3. Application Architecture**

* **3.1. Design Pattern:**
  + Layered architecture with a clear separation of concerns (e.g., presentation layer, business logic layer, data access layer).
* **3.2. Dependency Injection:**
  + If any dependency injection is used, it would likely be managed manually or with a basic IoC container like Unity for .NET.
* **3.3. Logging:**
  + Use Log4Net or Microsoft Enterprise Library for logging application events.
* **3.4. Exception Handling:**
  + Structured exception handling within the application, with try-catch blocks and logging of exceptions.

**4. User Interface Requirements**

* **4.1. Framework:**
  + Windows Forms (WinForms) for creating the user interface.
* **4.2. UI Components:**
  + Standard WinForms controls, with any necessary customizations coded manually.
* **4.3. Reporting:**
  + Crystal Reports 2013 for generating and printing reports from the application.

**5. Security Requirements**

* **5.1. Authentication:**
  + Custom authentication mechanisms, possibly leveraging Windows Authentication or a custom membership provider.
* **5.2. Data Encryption:**
  + Use basic encryption techniques, such as DPAPI (Data Protection API) for sensitive data.
* **5.3. Authorization:**
  + Role-based access control (RBAC) implemented through custom logic or using .NET’s built-in role management.
* **5.4. Secure Coding Practices:**
  + Follow OWASP guidelines from the 2013 period to prevent common vulnerabilities like SQL injection and XSS.

**6. Deployment Requirements**

* **6.1. Application Deployment:**
  + Use an MSI installer package created with Visual Studio Installer Projects or InstallShield.
* **6.2. Server Infrastructure:**
  + Windows Server 2012 or 2012 R2 for hosting SQL Server and other server-side components.
* **6.3. Load Balancing:**
  + Windows Network Load Balancing (NLB) for distributing traffic if the application is deployed in a web environment.
* **6.4. Application Monitoring:**
  + Basic server monitoring using Windows Performance Monitor (PerfMon) and event logs.

**7. Networking and Connectivity**

* **7.1. Network Requirements:**
  + The application must connect to SQL Server over a secure internal network, with proper firewall configurations.
* **7.2. Firewall Configuration:**
  + Ensure that the appropriate ports (e.g., 1433 for SQL Server) are open and secured on the firewall.
* **7.3. API Integration:**
  + Use SOAP-based web services if integrating with external systems, as this was more common during that period.

**8. Testing Requirements**

* **8.1. Unit Testing:**
  + Use MSTest for writing and executing unit tests within Visual Studio 2013.
* **8.2. Integration Testing:**
  + Basic integration testing through manual testing or custom scripts.
* **8.3. User Acceptance Testing (UAT):**
  + Set up a UAT environment similar to the production environment for final testing before deployment.
* **8.4. Load Testing:**
  + Use Visual Studio Load Test tools available in the 2013 edition for performing load testing.

**9. Documentation Requirements**

* **9.1. Code Documentation:**
  + Use XML comments in C# code for method and class-level documentation.
* **9.2. API Documentation:**
  + Manually create API documentation if SOAP services are used.
* **9.3. Deployment Documentation:**
  + Provide detailed deployment guides, including prerequisites, installation steps, and configuration instructions.

**10. Backup and Recovery Requirements**

* **10.1. Backup Strategy:**
  + Implement a full backup strategy with manual intervention for incremental backups.
* **10.2. Disaster Recovery:**
  + Documented disaster recovery plans with manual procedures for data recovery and system restoration.

**Architecture Design**

**1. Layered Architecture Overview**

The architecture will be divided into the following primary layers:

1. **Presentation Layer (UI Layer)**
2. **Business Logic Layer (BLL)**
3. **Data Access Layer (DAL)**
4. **Database Layer**

Each layer will have specific responsibilities, with the layers interacting in a hierarchical manner to process user requests and deliver responses.

**2. Layer Descriptions**

**2.1. Presentation Layer (UI Layer)**

* **Technology:** Windows Forms (WinForms)
* **Responsibilities:**
  + Provides the user interface for interacting with the application.
  + Handles user inputs, such as filling out purchase order forms and initiating approvals.
  + Communicates with the Business Logic Layer to process requests and display results.
  + Integrates with Crystal Reports for generating and printing reports.
* **Components:**
  + **Forms:** For various functions such as user authentication, purchase order creation, request review, and session management.
  + **Controls:** Standard WinForms controls (e.g., text boxes, dropdowns, buttons) and custom controls as needed.
  + **Reporting Module:** Embeds Crystal Reports for report generation and printing.

**2.2. Business Logic Layer (BLL)**

* **Technology:** C# with .NET Framework 4.5
* **Responsibilities:**
  + Contains the core business logic of the application.
  + Validates user inputs, processes business rules, and coordinates between the Presentation Layer and Data Access Layer.
  + Manages workflows such as purchase order submission, approval processes, and session management.
  + Handles security aspects such as role-based access control and authorization checks.
* **Components:**
  + **Business Services:** Classes that handle specific business operations (e.g., PurchaseOrderService, ApprovalService).
  + **Validation Logic:** Ensures data integrity and enforces business rules before data is processed.
  + **Security Manager:** Manages user roles and permissions, ensuring secure access to application functions.

**2.3. Data Access Layer (DAL)**

* **Technology:** ADO.NET
* **Responsibilities:**
  + Manages communication with the database, executing SQL queries and stored procedures.
  + Provides methods for CRUD (Create, Read, Update, Delete) operations on the database.
  + Ensures data is retrieved and stored efficiently, with proper error handling and transaction management.
* **Components:**
  + **Data Repositories:** Classes responsible for interacting with specific database tables (e.g., PurchaseOrderRepository, UserRepository).
  + **Database Context:** Manages connections to the SQL Server database, executing commands and returning results.
  + **Data Mappers:** Translates data between database rows and application objects.

**2.4. Database Layer**

* **Technology:** Microsoft SQL Server 2012/2014
* **Responsibilities:**
  + Stores all persistent data for the application, including user credentials, purchase orders, approval logs, and session details.
  + Ensures data integrity and security through appropriate constraints, relationships, and indexing.
  + Supports complex queries and reporting needs for insights and audit purposes.
* **Components:**
  + **Tables:** For storing entities like Users, Roles, PurchaseOrders, Approvals, and Sessions.
  + **Stored Procedures:** Encapsulates complex SQL logic for operations like order submission, approval workflows, and reporting.
  + **Views:** Provides pre-defined queries for reporting and UI purposes.
  + **Indexes:** Ensures fast retrieval of data for search and reporting functions.

**3. Workflow Example**

**Example: Purchase Order Approval Process**

1. **User Interaction:**
   * The user interacts with the PurchaseOrderReviewForm in the Presentation Layer.
   * They select a purchase order for review and decide to approve, modify, or reject it.
2. **Business Logic Processing:**
   * The PurchaseOrderService in the Business Logic Layer validates the user's action.
   * If approved, the service updates the order's status and initiates the next step in the approval process.
   * If rejected, the service updates the order's status and sends it back to the previous approver.
3. **Data Access:**
   * The PurchaseOrderRepository in the Data Access Layer updates the order record in the database.
   * The changes are committed to the SQL Server database, and the next step in the workflow is triggered.
4. **Database Update:**
   * The database layer records the updated approval status in the Approvals table.
   * The workflow continues based on the business logic defined in the stored procedures.

**4. Security Considerations**

* **Authentication:** Managed through the UI layer, where users sign in using Windows Authentication or custom login mechanisms.
* **Authorization:** Handled by the Business Logic Layer, ensuring that users have the appropriate roles and permissions to access or modify data.
* **Data Encryption:** Sensitive data is encrypted both at rest (in the database) and in transit (using SSL/TLS).

**5. Deployment Architecture**

* **Client-Server Model:** The application is deployed on client machines (Windows PCs) with the database hosted on a central SQL Server.
* **Database Server:** SQL Server is installed on a dedicated server, accessible over a secure network.
* **Client Machines:** Windows PCs with the CEA application installed, connected to the SQL Server via the internal network.

**1. Presentation Layer (UI Layer)**

This layer will have a simple console application to simulate the user interface.

using System;

using System.Collections.Generic;

namespace CEAApp {

class Program{

static void Main(string[] args) {

// Create an instance of the business logic layer

var purchaseOrderService = new PurchaseOrderService();

// Retrieve a list of purchase orders

List<PurchaseOrder> purchaseOrders = purchaseOrderService.GetPurchaseOrders();

// Display the purchase orders

foreach (var order in purchaseOrders) {

Console.WriteLine($"Order ID: {order.OrderId}, Title: {order.Title}, Amount: {order.Amount}");

}

Console.ReadLine();

}

}

}

**2. Business Logic Layer (BLL)**

This layer contains the business logic that interacts with the Data Access Layer.

using System.Collections.Generic;

namespace CEAApp {

public class PurchaseOrderService {

private readonly PurchaseOrderRepository \_repository;

public PurchaseOrderService() {

\_repository = new PurchaseOrderRepository(); // Dependency injection could be used in a more complex scenario

}

public List<PurchaseOrder> GetPurchaseOrders() {

// Business logic could be added here (e.g., filtering, validation)

return \_repository.GetAllPurchaseOrders();

}

}

}

**3. Data Access Layer (DAL)**

This layer is responsible for retrieving data from the database.

using System.Collections.Generic;

namespace CEAApp {

public class PurchaseOrderRepository {

public List<PurchaseOrder> GetAllPurchaseOrders() {

// Normally, database access would be here. We'll simulate with in-memory data.

return new List<PurchaseOrder> {

new PurchaseOrder { OrderId = 1, Title = "Office Supplies", Amount = 1500 },

new PurchaseOrder { OrderId = 2, Title = "IT Equipment", Amount = 45000 }

};

}

}

}

**4. Database Layer**

In a real application, this layer would interact with the SQL Server database using ADO.NET. Here, we are simulating the database layer with in-memory data.

namespace CEAApp {

public class PurchaseOrder {

public int OrderId { get; set; }

public string Title { get; set; }

public decimal Amount { get; set; }

}

}

**Explanation:**

* **Presentation Layer**: This is where the user interacts with the application. In this example, a console application is used to display purchase orders.
* **Business Logic Layer**: This layer handles the business logic. The PurchaseOrderService class interacts with the PurchaseOrderRepository to retrieve purchase orders.
* **Data Access Layer**: The PurchaseOrderRepository class simulates data retrieval. In a real-world scenario, this class would use ADO.NET to interact with the SQL Server database.
* **Database Layer**: The PurchaseOrder class represents a table in the database. In a real application, this would be tied to actual data in a SQL Server database.

**Database Design**

**1. Entity-Relationship Diagram (ERD)**

The ERD represents the relationships between different tables in the CEA application. Below is a simplified representation of key tables:

* **Tables:**
  + Users
  + Roles
  + PurchaseOrders
  + Approvals
* **Relationships:**
  + A User can have multiple PurchaseOrders.
  + A PurchaseOrder requires multiple Approvals.
  + A Role can be assigned to multiple Users.

The ERD would look something like this:

A screenshot of a computer screen

Description automatically generated

**2. Table Schema**

The following section describes the schema for each of the key tables:

**2.1. Users Table**

* **Table Name:** Users
* **Description:** Stores user information, including authentication and authorization details.

A screenshot of a computer

Description automatically generated

**2.2. Roles Table**

* **Table Name:** Roles
* **Description:** Stores the different roles that users can have.

A screenshot of a computer

Description automatically generated

**2.3. PurchaseOrders Table**

* **Table Name:** PurchaseOrders
* **Description:** Stores details about purchase orders created by users.

A black screen with white text

Description automatically generated

**2.4. Approvals Table**

* **Table Name:** Approvals
* **Description:** Tracks the approval status of each purchase order.

A black screen with white text

Description automatically generated

**3. Indexes and Relationships**

* **Indexes:**
  + Primary keys are automatically indexed.
  + Create an index on Username in the Users table to improve login performance.
  + Create an index on OrderId in the Approvals table for faster lookup of approvals.
* **Relationships:**
  + **Users to Roles:** A foreign key relationship exists between Users.RoleId and Roles.RoleId.
  + **Users to PurchaseOrders:** A foreign key relationship exists between PurchaseOrders.UserId and Users.UserId.
  + **PurchaseOrders to Approvals:** A foreign key relationship exists between Approvals.OrderId and PurchaseOrders.OrderId.

**4. Stored Procedures**

* **Create Purchase Order:**
  + A stored procedure to insert a new record into the PurchaseOrders table.

CREATE PROCEDURE CreatePurchaseOrder

@UserId INT,

@Title NVARCHAR(100),

@Amount DECIMAL(18,2)

AS

BEGIN

INSERT INTO PurchaseOrders (UserId, Title, Amount, DateCreated)

VALUES (@UserId, @Title, @Amount, GETDATE());

END

* **Get Purchase Orders:**
  + A stored procedure to retrieve all purchase orders, optionally filtered by UserId.

CREATE PROCEDURE GetPurchaseOrders

@UserId INT = NULL

AS

BEGIN

SELECT OrderId, UserId, Title, Amount, DateCreated

FROM PurchaseOrders

WHERE (@UserId IS NULL OR UserId = @UserId)

ORDER BY DateCreated DESC;

END

**5. Data Integrity and Constraints**

* **Foreign Key Constraints:**
  + Ensure that all foreign key relationships (e.g., between Users and Roles, PurchaseOrders and Users, etc.) are enforced by the database. This helps maintain referential integrity.
* **Check Constraints:**
  + Implement check constraints where applicable to ensure that data entered into the database meets specific conditions. For example:
    - Ensure that Amount in the PurchaseOrders table is always greater than 1 million pesos.
    - Ensure that Status in the Approvals table is limited to specific values (e.g., 'Approved', 'Rejected', 'Pending').

ALTER TABLE PurchaseOrders

ADD CONSTRAINT CHK\_Amount

CHECK (Amount >= 1000000);

ALTER TABLE Approvals

ADD CONSTRAINT CHK\_Status

CHECK (Status IN ('Approved', 'Rejected', 'Pending'));

**6. Transaction Management**

* **Stored Procedure Transactions:**
  + For operations that involve multiple steps, wrap the operations in a transaction to ensure atomicity. For example, when creating a purchase order and its associated approval records, you want to ensure that either all operations succeed or none at all.

CREATE PROCEDURE SubmitPurchaseOrder

@UserId INT,

@Title NVARCHAR(100),

@Amount DECIMAL(18,2)

AS

BEGIN

BEGIN TRANSACTION

BEGIN TRY

-- Insert Purchase Order

INSERT INTO PurchaseOrders (UserId, Title, Amount, DateCreated)

VALUES (@UserId, @Title, @Amount, GETDATE());

DECLARE @OrderId INT = SCOPE\_IDENTITY();

-- Insert Initial Approval Record

INSERT INTO Approvals (OrderId, ApproverId, Status)

VALUES (@OrderId, @UserId, 'Pending');

COMMIT TRANSACTION

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION

THROW;

END CATCH

END

**7. Auditing and Logging**

* **Audit Tables:**
  + Create audit tables to track changes to critical data, such as modifications to purchase orders or user roles. This provides a historical record for compliance and debugging purposes.

CREATE TABLE AuditLogs (

AuditId INT IDENTITY PRIMARY KEY,

TableName NVARCHAR(100),

Operation NVARCHAR(50),

RecordId INT,

ChangedBy INT,

DateChanged DATETIME,

OldValue NVARCHAR(MAX),

NewValue NVARCHAR(MAX)

);

* **Triggers:**
  + Use triggers to automatically log changes to important tables into an audit log. For example, logging changes to the PurchaseOrders table.

sql

Copy code

CREATE TRIGGER trg\_PurchaseOrderAudit

ON PurchaseOrders

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

-- Example logic to insert into AuditLogs

END

**8. Indexes and Performance Tuning**

* **Index Strategy:**
  + Beyond primary keys, consider creating composite indexes on frequently queried columns or columns involved in JOIN operations to improve performance.

CREATE INDEX IX\_PurchaseOrders\_UserId\_DateCreated

ON PurchaseOrders(UserId, DateCreated);

* **Index Maintenance:**
  + Document how indexes will be maintained, such as rebuilding or reorganizing indexes periodically to ensure optimal performance.

**9. Backup and Recovery**

* **Backup Strategy:**
  + Document the backup strategy, including frequency (e.g., full backup daily, differential backups hourly), retention period, and off-site storage.
* **Point-in-Time Recovery:**
  + Ensure that the database is configured for point-in-time recovery to allow restoration to a specific moment in case of data corruption or loss.

**10. Security Considerations**

* **User Roles and Permissions:**
  + Clearly define and document the roles and permissions within SQL Server. For example:
    - DB\_DataReader: Users who need to read data.
    - DB\_DataWriter: Users who need to insert, update, or delete data.
    - DB\_Owner: Users with full control over the database.
* **Encryption:**
  + Ensure that sensitive columns are encrypted using SQL Server’s encryption capabilities, and document the encryption keys used.

**11. Future Considerations**

* **Scalability:**
  + Consider potential future growth. Document considerations for partitioning tables, scaling the database, and distributing load across multiple servers.
* **Archiving Data:**
  + Plan for archiving old data. For instance, consider archiving or purging purchase orders older than a certain period to maintain performance.

**12. Documentation of Database Access**

* **Connection Strings:**
  + Include sample connection strings in the documentation, detailing how the application should connect to the database securely.

Server=myServerAddress;Database=myDataBase;User Id=myUsername;Password=myPassword;

* **Database User Accounts:**
  + Document which accounts the application will use to connect to the database, along with their permissions and roles.

**Class Diagram Overview**

The diagram will include the following classes:

1. **User**
2. **Role**
3. **PurchaseOrder**
4. **Approval**
5. **PurchaseOrderService**
6. **PurchaseOrderRepository**

**Class Diagram Description**

* **User Class**
  + **Attributes:**
    - UserId: int
    - Username: string
    - PasswordHash: string
    - Role: Role
  + **Methods:**
    - Authenticate(password: string): bool
* **Role Class**
  + **Attributes:**
    - RoleId: int
    - RoleName: string
  + **Methods:**
    - (None)
* **PurchaseOrder Class**
  + **Attributes:**
    - OrderId: int
    - User: User
    - Title: string
    - Amount: decimal
    - DateCreated: DateTime
  + **Methods:**
    - Submit(): void
    - Approve(user: User): void
    - Reject(user: User): void
* **Approval Class**
  + **Attributes:**
    - ApprovalId: int
    - PurchaseOrder: PurchaseOrder
    - Approver: User
    - Status: string
    - DateApproved: DateTime
  + **Methods:**
    - SetStatus(status: string): void
* **PurchaseOrderService Class**
  + **Attributes:**
    - (None)
  + **Methods:**
    - GetPurchaseOrders(): List<PurchaseOrder>
    - SubmitPurchaseOrder(order: PurchaseOrder): void
* **PurchaseOrderRepository Class**
  + **Attributes:**
    - (None)
  + **Methods:**
    - GetAllPurchaseOrders(): List<PurchaseOrder>
    - Save(order: PurchaseOrder): void

**Simplified Class Diagram (Text Representation)**

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Explanation:**

* **User and Role Classes**: The User class is associated with a Role class, representing the role-based access control in the system. A User can have a single Role, and the Role class defines the specific permissions or access level.
* **PurchaseOrder Class**: The PurchaseOrder class represents the purchase order entity, with methods for submitting, approving, and rejecting the order. Each purchase order is associated with a User who created it.
* **Approval Class**: The Approval class is used to track the approval status of each purchase order. It is associated with a PurchaseOrder and a User who approved or rejected the order.
* **PurchaseOrderService Class**: This service class handles the business logic related to purchase orders, such as retrieving orders and submitting new orders.
* **PurchaseOrderRepository Class**: This class interacts with the database, encapsulating the data access logic for purchase orders.

**UX/UI Design Documentation**

**1. User Personas**

**User Persona 1: Luis García – Purchasing Employee**

* **Age:** 28
* **Role:** Purchasing Employee
* **Experience Level:** Intermediate
* **Location:** Mexico City, Mexico
* **Background:**
  + Luis has been working in the purchasing department for 5 years. He handles the creation of purchase orders, ensuring all necessary documentation is complete before submitting for approval.
  + He is familiar with the company's processes but sometimes struggles with the current manual system, leading to errors and delays.
* **Goals:**
  + Quickly create and submit purchase orders with minimal errors.
  + Easily track the status of submitted orders.
  + Ensure that all documentation, including vendor comparisons, is uploaded correctly.
* **Frustrations:**
  + Manual processes that are prone to errors and delays.
  + Difficulty in tracking the status of purchase orders.
  + Confusion due to unclear UI elements in the existing system.
* **Motivations:**
  + Wants to improve efficiency and accuracy in order processing.
  + Aims to reduce the back-and-forth communication with supervisors for clarifications.
* **Technical Skills:**
  + Comfortable with basic computer applications (Microsoft Office, Internet browsing).
  + Limited experience with more complex systems or software.
* **Needs from the Application:**
  + An intuitive, easy-to-navigate interface for creating and submitting purchase orders.
  + Clear instructions and validations to minimize errors.
  + Ability to quickly upload and manage supporting documents like vendor comparisons.

**User Persona 2: Marta Rivera – Purchasing Supervisor**

* **Age:** 35
* **Role:** Supervisor (Purchasing Department)
* **Experience Level:** Advanced
* **Location:** Monterrey, Mexico
* **Background:**
  + Marta has been with the company for over 10 years, rising through the ranks to her current supervisory role. She oversees the work of purchasing employees and is responsible for the first level of approval for purchase orders.
  + Marta is very detail-oriented and ensures that all purchase orders comply with company policies before moving them forward in the approval process.
* **Goals:**
  + Efficiently review and approve purchase orders.
  + Ensure compliance with company policies and procurement standards.
  + Minimize the time spent on administrative tasks to focus on strategic decision-making.
* **Frustrations:**
  + Wastes time correcting errors made by purchasing employees.
  + Limited visibility into the overall approval workflow in the current system.
  + Difficulty in managing and organizing multiple purchase orders at once.
* **Motivations:**
  + Wants to streamline the approval process to improve departmental efficiency.
  + Motivated by ensuring that all purchases meet company standards and deliver value.
* **Technical Skills:**
  + Proficient with business applications and ERP systems.
  + Comfortable using complex software and tools.
* **Needs from the Application:**
  + A dashboard that provides an overview of pending approvals.
  + Tools to quickly review, approve, or send back purchase orders for revision.
  + The ability to track the status of purchase orders across different stages of the workflow.

**User Persona 3: Carlos López – Purchasing Manager**

* **Age:** 45
* **Role:** Manager (Purchasing Department)
* **Experience Level:** Expert
* **Location:** Guadalajara, Mexico
* **Background:**
  + Carlos is a seasoned professional with 20 years of experience in purchasing and procurement. He manages the overall operations of the purchasing department and ensures that procurement strategies align with company goals.
  + Carlos regularly interacts with senior management and the general direction to approve large purchases.
* **Goals:**
  + Oversee and approve high-value purchase orders.
  + Analyze procurement data to identify trends and make strategic decisions.
  + Ensure that the purchasing process is efficient and cost-effective.
* **Frustrations:**
  + Delays in the approval process due to bottlenecks at lower levels.
  + Lack of insightful reports that provide a clear view of departmental performance.
  + The current system does not support his need for detailed data analysis.
* **Motivations:**
  + Streamlining operations to reduce costs and improve procurement efficiency.
  + Driving strategic decisions based on accurate data and reports.
* **Technical Skills:**
  + Highly proficient with ERP systems, data analysis tools, and advanced business applications.
  + Familiar with generating and interpreting complex reports.
* **Needs from the Application:**
  + Advanced reporting tools that provide insights into procurement activities.
  + A seamless interface for approving large purchase orders, with clear visibility into the history and status of each order.
  + The ability to manage and organize purchase orders at a strategic level.

**User Persona 4: Ana Martínez – Director of Purchasing**

* **Age:** 50
* **Role:** Director of Purchasing
* **Experience Level:** Executive
* **Location:** Mexico City, Mexico
* **Background:**
  + Ana is an executive with extensive experience in corporate procurement and strategic sourcing. She leads the purchasing department and works closely with the general direction to ensure that all major purchases align with the company's financial and operational goals.
  + Ana's role requires her to make high-level decisions and provide final approval for large purchase orders.
* **Goals:**
  + Ensure that all major purchase decisions are aligned with the company’s strategic objectives.
  + Maintain oversight of the purchasing department’s performance.
  + Facilitate the approval of large purchase orders efficiently.
* **Frustrations:**
  + Time-consuming manual approval processes.
  + Lack of real-time visibility into procurement activities and approval statuses.
  + Inconsistent reporting and data quality from the current system.
* **Motivations:**
  + Streamlining the approval process to focus more on strategic initiatives.
  + Ensuring that procurement decisions deliver maximum value to the company.
* **Technical Skills:**
  + Proficient with enterprise-level business systems and financial reporting tools.
  + Comfortable with technology but prefers intuitive and streamlined interfaces.
* **Needs from the Application:**
  + A high-level overview dashboard showing the status of all major purchase orders.
  + Tools to quickly approve or reject purchase orders with access to all necessary information.
  + Detailed and customizable reports that provide insights into procurement activities and spending.

**2. User Journey Maps**

**User Journey Map: Luis García – Purchasing Employee**

**Goal: Create and Submit a Purchase Order**

**1. Sign In**

* **Step:** Luis navigates to the CEA application and enters his username and password.
* **Interaction:** Input fields for username and password.
* **Pain Points:** If login fails, Luis may face frustration due to unclear error messages.
* **Opportunities:** Provide clear error messages and a "Forgot Password" option to reduce login-related issues.

**2. Access Purchase Order Creation Screen**

* **Step:** After logging in, Luis selects "Create Purchase Order" from the main menu.
* **Interaction:** Main navigation menu.
* **Pain Points:** Difficulty in locating the purchase order creation option due to a cluttered or confusing UI.
* **Opportunities:** Ensure the menu is intuitive with clear labeling and a logical order of options.

**3. Fill Out Purchase Order Form**

* **Step:** Luis fills in the mandatory fields, including order title, amount, and vendor details.
* **Interaction:** Form fields for entering purchase order details.
* **Pain Points:** Confusion over which fields are mandatory, leading to incomplete submissions.
* **Opportunities:** Use visual cues (e.g., asterisks, tooltips) to clearly indicate mandatory fields.

**4. Upload Vendor Comparison Document**

* **Step:** Luis uploads an image or document comparing vendor prices.
* **Interaction:** File upload field.
* **Pain Points:** File upload issues (e.g., unsupported formats, size limitations).
* **Opportunities:** Provide clear guidelines on acceptable file types and sizes, and offer feedback on successful uploads.

**5. Submit Purchase Order**

* **Step:** After reviewing the form, Luis clicks "Submit" to send the purchase order for approval.
* **Interaction:** Submit button, confirmation dialog.
* **Pain Points:** Uncertainty if the order was successfully submitted due to lack of confirmation.
* **Opportunities:** Display a clear confirmation message and provide a summary of the submitted order.

**6. Track Purchase Order Status**

* **Step:** Luis checks the status of his submitted orders from the dashboard.
* **Interaction:** Dashboard with order status overview.
* **Pain Points:** Difficulty in finding or interpreting the status of pending orders.
* **Opportunities:** Use clear status indicators (e.g., color coding, progress bars) and offer sorting/filtering options.

**User Journey Map: Marta Rivera – Purchasing Supervisor**

**Goal: Review and Approve a Purchase Order**

**1. Sign In**

* **Step:** Marta logs in using her credentials.
* **Interaction:** Input fields for username and password.
* **Pain Points:** Same as for Luis; ensuring smooth login is key.
* **Opportunities:** Consider a "Keep me signed in" option for frequent users like Marta.

**2. Access Pending Purchase Orders**

* **Step:** After logging in, Marta navigates to the "Pending Approvals" section.
* **Interaction:** Main navigation menu, dashboard view of pending approvals.
* **Pain Points:** Overloaded or cluttered dashboard making it hard to prioritize.
* **Opportunities:** Allow filtering and sorting by urgency, value, or department to help Marta manage her workload.

**3. Review Purchase Order Details**

* **Step:** Marta selects a purchase order to review the details, including the uploaded vendor comparison.
* **Interaction:** Detailed view of the purchase order, including all fields and attachments.
* **Pain Points:** Inability to quickly access key details, or poor layout leading to confusion.
* **Opportunities:** Highlight key information and provide an option to zoom in on or download attachments for a closer look.

**4. Approve, Modify, or Reject the Purchase Order**

* **Step:** After reviewing, Marta decides whether to approve, modify, or reject the order.
* **Interaction:** Approve/Modify/Reject buttons, form fields for modifications.
* **Pain Points:** Unclear options or lack of feedback after action is taken.
* **Opportunities:** Ensure that each option is clearly labeled and that Marta receives immediate confirmation and feedback for her action.

**5. Provide Feedback if Rejected**

* **Step:** If Marta rejects the order, she provides feedback to the purchasing employee.
* **Interaction:** Comment box or feedback form.
* **Pain Points:** Limited space or character limits for providing detailed feedback.
* **Opportunities:** Offer sufficient space for detailed feedback and allow attachments if needed.

**6. Monitor Order Progress**

* **Step:** Marta monitors the progress of the orders she has approved.
* **Interaction:** Dashboard view with status updates.
* **Pain Points:** Lack of real-time updates or notifications on status changes.
* **Opportunities:** Implement real-time notifications or a system-generated summary email for status updates.

**User Journey Map: Carlos López – Purchasing Manager**

**Goal: Analyze Reports and Approve High-Value Orders**

**1. Sign In**

* **Step:** Carlos logs in to access the application.
* **Interaction:** Input fields for username and password.
* **Pain Points:** Similar to other users; ensuring smooth login is crucial.
* **Opportunities:** Implement single sign-on (SSO) for a more seamless experience.

**2. Access Reports Section**

* **Step:** Carlos navigates to the "Reports" section from the main menu.
* **Interaction:** Main navigation menu, reporting dashboard.
* **Pain Points:** Difficulty in finding relevant reports due to a complex or disorganized menu.
* **Opportunities:** Organize reports by category (e.g., monthly, department-wise) and offer a search feature.

**3. Generate and Analyze Reports**

* **Step:** Carlos selects the parameters to generate a report (e.g., date range, department) and analyzes the results.
* **Interaction:** Reporting tool with filters and parameters.
* **Pain Points:** Long loading times for generating reports or lack of customization options.
* **Opportunities:** Optimize report generation times and provide a variety of filters and customization options.

**4. Review High-Value Orders Pending Approval**

* **Step:** Carlos reviews the high-value purchase orders pending his approval.
* **Interaction:** Dashboard with a focus on high-value orders.
* **Pain Points:** Difficulty in quickly accessing high-priority items or missing critical information.
* **Opportunities:** Highlight high-value orders and ensure all necessary information is easily accessible from a single screen.

**5. Approve or Request Further Review**

* **Step:** Carlos decides whether to approve the order or request further review.
* **Interaction:** Approve button, option to flag for further review.
* **Pain Points:** Uncertainty about the implications of approval or lack of tools for requesting additional information.
* **Opportunities:** Provide clear guidance on the approval process and offer tools to request additional data or clarification from other users.

**User Journey Map: Ana Martínez – Director of Purchasing**

**Goal: Provide Final Approval for Major Purchases**

**1. Sign In**

* **Step:** Ana logs in to the application.
* **Interaction:** Input fields for username and password.
* **Pain Points:** Ensuring a secure and straightforward login experience, especially for high-level users.
* **Opportunities:** Consider implementing biometric authentication for executives like Ana.

**2. Access High-Level Dashboard**

* **Step:** Ana views the high-level dashboard that shows an overview of all major purchases.
* **Interaction:** Dashboard with aggregated data and status summaries.
* **Pain Points:** Information overload or irrelevant data cluttering the view.
* **Opportunities:** Customize the dashboard to show only the most critical information relevant to Ana’s role.

**3. Review Major Purchase Orders**

* **Step:** Ana selects major purchase orders that require her approval.
* **Interaction:** Detailed view of high-value purchase orders.
* **Pain Points:** Lack of clarity or missing data in the order details.
* **Opportunities:** Ensure that all necessary information is presented clearly and concisely, with links to supporting documents.

**4. Make Final Approval Decision**

* **Step:** Ana approves or rejects the purchase order based on the provided information.
* **Interaction:** Approve/Reject buttons, with an option to leave comments.
* **Pain Points:** Potential hesitancy due to lack of confidence in the presented data.
* **Opportunities:** Offer a summary of the approval history and ensure that the data is presented in a way that instills confidence in the decision-making process.

**5. Confirm and Close**

* **Step:** Ana confirms her decision, which finalizes the approval process.
* **Interaction:** Confirmation dialog, final approval.
* **Pain Points:** Uncertainty about whether the process has been completed successfully.
* **Opportunities:** Provide a clear confirmation message and summary, and offer an option to print or save the decision for records.

**3. Wireframes and Mockups**

**1. Login Screen Wireframe**

**Purpose:**

To allow users to securely log into the application.

**Key Elements:**

* **Username Field:** A text input field labeled "Username."
* **Password Field:** A password input field labeled "Password."
* **Sign In Button:** A primary button labeled "Sign In."
* **Forgot Password Link:** A text link beneath the password field, labeled "Forgot Password?"
* **Error Message Area:** A space to display error messages (e.g., "Incorrect username or password").
* **Company Logo:** Positioned at the top center of the screen.

**Layout:**

* **Top:** Company logo centered.
* **Middle:** Stacked input fields for "Username" and "Password," followed by the "Sign In" button.
* **Bottom:** A link for "Forgot Password?" and space for any error messages.

**Interactions:**

* When the user clicks "Sign In," the system checks credentials.
* If incorrect, an error message appears in the designated area.
* Clicking "Forgot Password?" navigates to a password recovery screen.

**2. Dashboard Wireframe**

**Purpose:**

To provide users with an overview of their tasks, including pending purchase orders and approvals.

**Key Elements:**

* **Main Navigation Menu:** A sidebar or top bar with links to "Dashboard," "Create Purchase Order," "Pending Approvals," "Reports," and "Settings."
* **Welcome Message:** A greeting at the top, e.g., "Welcome, Luis!"
* **Pending Tasks Section:** A list of pending purchase orders or approvals.
* **Quick Links:** Buttons for common actions, such as "Create New Purchase Order" and "View All Orders."
* **Search Bar:** A search field to quickly find orders or tasks.
* **Notifications Area:** A bell icon or similar indicator for notifications.

**Layout:**

* **Left Sidebar (or Top Bar):** Main navigation links.
* **Top:** Welcome message and search bar.
* **Middle:** Pending tasks section with a list format.
* **Right (or Bottom):** Quick links for frequently used actions.

**Interactions:**

* Clicking on a task in the "Pending Tasks" section opens the detailed view of that task.
* The search bar allows filtering of tasks by order number, title, etc.
* The navigation menu provides access to different sections of the application.

**3. Create Purchase Order Screen Wireframe**

**Purpose:**

To allow users to create a new purchase order by filling in necessary details.

**Key Elements:**

* **Form Fields:** Input fields for "Order Title," "Amount," "Vendor," "Request Type," "Contract Number," etc.
* **Upload Section:** A button to upload the vendor comparison document.
* **Save Draft Button:** A secondary button labeled "Save Draft."
* **Submit Button:** A primary button labeled "Submit Purchase Order."
* **Cancel Button:** A link or button to cancel the operation and return to the dashboard.
* **Form Validation Messages:** Inline messages for form validation (e.g., "This field is required").

**Layout:**

* **Top:** Form fields stacked vertically, with labels aligned to the left.
* **Middle:** The upload section, prominently placed to ensure users don't miss it.
* **Bottom:** Action buttons (Save Draft, Submit, Cancel) aligned to the right or center.

**Interactions:**

* The "Submit" button triggers validation of all required fields and uploads.
* The "Save Draft" button saves the current form state for later completion.
* The "Cancel" button discards the form and returns to the previous screen.

**4. Pending Approvals Screen Wireframe**

**Purpose:**

To allow supervisors and managers to review and approve or reject purchase orders.

**Key Elements:**

* **List of Purchase Orders:** A table or list format displaying orders with columns like "Order ID," "Title," "Amount," "Status," and "Actions."
* **Filter and Sort Options:** Dropdowns or buttons to filter orders by status, department, date, etc.
* **Approve/Reject/Modify Buttons:** Buttons for each order to take action.
* **Order Details Button:** A link or button to view more detailed information about each order.

**Layout:**

* **Top:** Filter and sort options aligned horizontally.
* **Middle:** List of purchase orders with inline actions (Approve/Reject/Modify).
* **Bottom:** Pagination controls or a "Load More" button if there are many orders.

**Interactions:**

* Clicking "Approve" changes the order status and moves it to the next step in the workflow.
* "Reject" prompts for feedback and returns the order to the previous step.
* "Modify" allows the supervisor to make changes to the order details before resubmission.

**5. Reports Screen Wireframe**

**Purpose:**

To allow users to generate and view reports on purchase orders and approvals.

**Key Elements:**

* **Report Filters:** Dropdowns, date pickers, and checkboxes to select report parameters.
* **Generate Report Button:** A primary button labeled "Generate Report."
* **Report Viewer:** An embedded viewer or link to download the generated report.
* **Saved Reports:** A list or dropdown to access previously generated reports.

**Layout:**

* **Top:** Filters and parameters for report generation.
* **Middle:** The report viewer area where the generated report is displayed.
* **Bottom:** Links or buttons to access saved reports or export options.

**Interactions:**

* The "Generate Report" button runs the selected report with the chosen parameters.
* Users can save the report for later access or export it in different formats (PDF, Excel).

**6. Purchase Order Details Screen Wireframe**

**Purpose:**

To provide a detailed view of a specific purchase order, including all entered data and the approval history.

**Key Elements:**

* **Order Summary:** A section at the top summarizing the key details (Order ID, Title, Amount, Date Created, etc.).
* **Detailed Fields:** All fields filled in during the creation of the purchase order (e.g., Vendor, Contract Number, Request Type).
* **Uploaded Documents:** A list or thumbnails of uploaded documents with options to view or download them.
* **Approval History:** A timeline or list showing the status of approvals, including who approved/rejected, the date, and any comments.
* **Back Button:** A button to return to the previous screen, such as the dashboard or pending approvals list.

**Layout:**

* **Top:** Order Summary with key information displayed prominently.
* **Middle Left:** Detailed fields, laid out in a form-style manner.
* **Middle Right:** Uploaded documents section.
* **Bottom:** Approval History presented in a timeline or table format.
* **Footer:** Back button to navigate to the previous screen.

**Interactions:**

* Clicking on an uploaded document opens it in a viewer or downloads it.
* Hovering over approval history entries may show additional details (e.g., comments).

**7. Session Management Screen Wireframe**

**Purpose:**

To allow users to create, manage, and group purchase orders into sessions for presentation to the general direction.

**Key Elements:**

* **Create New Session Button:** A primary button labeled "Create New Session."
* **List of Sessions:** A table or list showing existing sessions, with columns for Session ID, Title, Number of Orders, Status, and Actions.
* **Session Details:** When a session is selected, details appear on the right, including the list of purchase orders in the session and their status.
* **Add/Remove Orders:** Controls to add or remove purchase orders from a session.
* **Submit Session Button:** A button to submit the session for approval by the general direction.

**Layout:**

* **Left:** List of existing sessions with a "Create New Session" button above it.
* **Right:** Detailed view of the selected session, with options to manage the included orders.
* **Bottom Right:** Submit Session button prominently displayed.

**Interactions:**

* Clicking "Create New Session" opens a form to input session details (title, description).
* Selecting a session from the list shows detailed information and management options.
* The "Submit Session" button triggers the process of presenting the session to the general direction.

**8. Session Details Screen Wireframe**

**Purpose:**

To review the details of a session, including all grouped purchase orders, before submission to the general direction.

**Key Elements:**

* **Session Overview:** A section summarizing the session (Session ID, Title, Date Created, etc.).
* **Grouped Purchase Orders:** A list of all purchase orders included in the session, with their statuses.
* **Approval Status:** A summary of the session’s current status in the approval workflow.
* **Edit Session Button:** A button to edit the session, allowing for adding/removing orders or modifying session details.
* **Submit for Final Approval Button:** A primary button to submit the session for final approval.

**Layout:**

* **Top:** Session Overview section.
* **Middle:** List of grouped purchase orders, displayed in a table format.
* **Bottom Left:** Edit Session button.
* **Bottom Right:** Submit for Final Approval button.

**Interactions:**

* Clicking "Edit Session" allows modification of session details and included orders.
* The "Submit for Final Approval" button moves the session to the final approval stage.

**9. Crystal Report Generation Screen Wireframe**

**Purpose:**

To generate and print reports summarizing purchase orders, for physical signatures by the general directors.

**Key Elements:**

* **Report Template Selection:** Dropdown or list to select the type of report to generate (e.g., Summary Report, Detailed Report).
* **Report Filters:** Options to filter the data to be included in the report (e.g., Date Range, Department).
* **Generate Report Button:** A primary button labeled "Generate Report."
* **Report Preview:** A section to preview the generated report before printing.
* **Print Button:** A primary button labeled "Print Report."
* **Export Options:** Buttons to export the report as PDF, Excel, or Word.

**Layout:**

* **Top:** Report template selection and filter options.
* **Middle:** Report preview area, showing the generated report.
* **Bottom Left:** Export options (PDF, Excel, Word).
* **Bottom Right:** Generate Report and Print Report buttons.

**Interactions:**

* Selecting a report template updates the preview area with the chosen report format.
* The "Generate Report" button refreshes the preview with the selected filters.
* The "Print Report" button sends the report to the printer, formatted for physical signatures.

**10. Session Closure Screen Wireframe**

**Purpose:**

To allow users to close sessions and mark cases as approved or rejected by the general direction.

**Key Elements:**

* **Session Selection:** A dropdown or list to select an open session.
* **Session Overview:** Display of selected session details, including the list of purchase orders and their statuses.
* **Final Approval Status:** A section to view and update the final approval status for each order in the session.
* **Close Session Button:** A primary button labeled "Close Session" to finalize the session.

**Layout:**

* **Top Left:** Session Selection dropdown.
* **Top Right:** Session Overview with detailed session information.
* **Middle:** Final Approval Status, with options to approve or reject each order.
* **Bottom Right:** Close Session button.

**Interactions:**

* Selecting a session shows the detailed view and current status of each order.
* The "Close Session" button finalizes the session, updating the status of all included orders.

**11. Settings Screen Wireframe**

**Purpose:**

To manage user settings, roles, permissions, and application configurations.

**Key Elements:**

* **User Management:** Section for adding, editing, or removing users, assigning roles, and managing permissions.
* **Application Settings:** Options for configuring application-wide settings, such as time zones, default currency, etc.
* **Audit Logs:** Access to view and export logs of user actions and system events.
* **Save Settings Button:** A button to save any changes made on this screen.

**Layout:**

* **Left Sidebar:** Navigation links for different settings categories (User Management, Application Settings, Audit Logs).
* **Right:** Detailed settings options based on the selected category.
* **Bottom Right:** Save Settings button.

**Interactions:**

* Navigating through the sidebar changes the settings options displayed.
* The "Save Settings" button confirms and applies changes made on the screen.

**12. User Profile Screen**

**Purpose:**

To allow users to view and update their personal information, such as contact details, password, and preferences.

**Key Elements:**

* **Personal Information:** Fields for name, email, phone number, etc.
* **Change Password:** A section to change the user’s password, including fields for the current password, new password, and confirmation.
* **Preferences:** Settings like notification preferences, language, and time zone.
* **Save Changes Button:** A button to save any updates made by the user.

**Interactions:**

* Users can update their information and save changes.
* Password changes require entering the current password for security.

**13. Notification Center Screen**

**Purpose:**

To display system notifications and alerts, such as pending approvals, order updates, and system messages.

**Key Elements:**

* **Notification List:** A list of notifications, each with a title, short description, and timestamp.
* **Filter/Sort Options:** Options to filter notifications by type (e.g., approvals, system alerts) or sort by date.
* **Mark as Read:** Option to mark notifications as read, either individually or all at once.
* **Detailed View:** Clicking a notification opens a more detailed view or navigates to the relevant section.

**Interactions:**

* Users can click on notifications to take action or view more details.
* Filter and sort options help manage and organize notifications.

**14. Audit Log Viewer Screen**

**Purpose:**

To allow administrators and authorized users to view detailed logs of user actions and system events for auditing purposes.

**Key Elements:**

* **Log List:** A table displaying logs with columns for date/time, user, action performed, and details.
* **Filter Options:** Filters for date range, user, action type, etc.
* **Search Bar:** A search field to find specific events or actions quickly.
* **Export Logs:** Option to export the logs to CSV or PDF for further analysis.

**Interactions:**

* Users can filter and search through logs to investigate specific actions.
* Export functionality allows saving logs for compliance or reporting purposes.

**15. Help/Support Center Screen**

**Purpose:**

To provide users with access to help documentation, FAQs, and support resources.

**Key Elements:**

* **Search Bar:** A search field to find help articles or FAQs.
* **FAQ List:** A list of frequently asked questions, categorized by topic.
* **Help Articles:** Links to detailed help articles or guides.
* **Contact Support:** Options to contact support via email, phone, or a support ticket system.
* **Tutorials/Guides:** Access to video tutorials or step-by-step guides.

**Interactions:**

* Users can search for help topics or browse through categorized FAQs.
* Contact support options provide a direct way to get assistance.

**16. Admin Dashboard Screen**

**Purpose:**

To give administrators an overview of system performance, user activity, and other administrative tasks.

**Key Elements:**

* **System Metrics:** Widgets or charts showing system usage, performance, and error rates.
* **User Activity:** A section showing recent user logins, most active users, and any suspicious activity.
* **Quick Links:** Access to administrative tasks like user management, system settings, and audit logs.
* **Notifications:** Alerts for system updates, maintenance, or critical issues.

**Interactions:**

* Admins can view metrics and click through to more detailed reports or logs.
* Quick links provide direct access to critical administrative functions.

**17. Error Handling Screen**

**Purpose:**

To inform users of system errors and provide guidance on how to resolve them or report them to support.

**Key Elements:**

* **Error Message:** A clear and concise message explaining what went wrong.
* **Error Code:** A specific error code or ID that can be reported to support.
* **Suggested Actions:** Steps the user can take to resolve the issue or mitigate its impact.
* **Contact Support:** Direct links to contact support if the issue cannot be resolved by the user.

**Interactions:**

* Users can follow suggested actions or report the issue to support.
* The screen should be designed to be as helpful as possible, minimizing user frustration during errors.

**18. Confirmation/Success Screens**

**Purpose:**

To confirm that a user’s action was successful, such as submitting a purchase order or saving changes.

**Key Elements:**

* **Confirmation Message:** A message confirming the action, such as "Your purchase order has been successfully submitted."
* **Next Steps:** Suggestions for what the user might want to do next, like "Return to Dashboard" or "View Purchase Order."
* **Close or Continue Button:** A button to close the confirmation or continue to the next step.

**Interactions:**

* Users are reassured that their action was completed successfully and given options for what to do next.

**19. User Onboarding/Tutorial Screen**

**Purpose:**

To guide new users through the application’s key features and workflows.

**Key Elements:**

* **Introduction:** A brief welcome message explaining the purpose of the onboarding.
* **Step-by-Step Guide:** Interactive steps that highlight key features or workflows, with explanations.
* **Skip Tutorial Option:** An option to skip the onboarding process.
* **Completion Message:** A message indicating the tutorial is complete, with links to start using the application.

**Interactions:**

* Users follow along with the tutorial, which highlights and explains different parts of the UI.
* The option to skip or revisit the tutorial is provided for flexibility.

**20. User Feedback Screen**

**Purpose:**

To allow users to provide feedback on the application, report bugs, or suggest improvements.

**Key Elements:**

* **Feedback Form:** Fields for users to describe their feedback or issue.
* **Rating System:** An optional rating system (e.g., 1 to 5 stars) to gauge user satisfaction.
* **Submit Button:** A button to submit the feedback.
* **Acknowledgment Message:** A confirmation that the feedback has been received.

**Interactions:**

* Users can submit feedback directly from the application, helping to improve future versions.

**21. Search Results Screen**

**Purpose:**

To display the results of a search query, such as searching for purchase orders or sessions.

**Key Elements:**

* **Search Bar:** A search field at the top to refine the query.
* **Results List:** A list or table displaying search results with relevant details.
* **Filter/Sort Options:** Options to filter or sort results to find the most relevant entries.
* **Details Link:** A link or button to view more details about each result.

**Interactions:**

* Users can refine their search query or apply filters to narrow down results.
* Clicking on a result takes the user to the detailed view of the selected item.

**4. UI Components Library**

* **UI Elements:**

**1. Buttons**

* **Primary Button:**
  + **Usage:** For primary actions like "Submit," "Save," or "Next."
  + **Appearance:** Solid color background (e.g., blue), white text, slight rounded corners.
  + **States:**
    - **Default:** Blue background, white text.
    - **Hover:** Darker blue background, white text.
    - **Active:** Even darker blue background, slightly inset to indicate it’s being clicked.
    - **Disabled:** Light gray background, gray text, no hover effect.
* **Secondary Button:**
  + **Usage:** For secondary actions like "Cancel," "Back," or "Edit."
  + **Appearance:** Transparent or light-colored background, blue text, border with rounded corners.
  + **States:**
    - **Default:** Transparent background, blue text.
    - **Hover:** Light blue background, blue text.
    - **Active:** Darker border, inset effect.
    - **Disabled:** Transparent background, gray text, light gray border.
* **Icon Button:**
  + **Usage:** For actions that are represented by icons (e.g., search, settings).
  + **Appearance:** Icon-only, no text. Usually a simple color change on hover.
  + **States:**
    - **Default:** Gray icon.
    - **Hover:** Blue icon.
    - **Active:** Darker blue icon.

**2. Form Elements**

* **Text Input:**
  + **Usage:** For entering text, such as usernames, titles, or amounts.
  + **Appearance:** White background, gray border, light padding.
  + **States:**
    - **Default:** Light gray border.
    - **Focus:** Blue border, slight shadow.
    - **Error:** Red border, with an error message below.
    - **Disabled:** Light gray background, dark gray border, no focus state.
* **Password Input:**
  + **Usage:** For entering passwords.
  + **Appearance:** Same as Text Input, with password masking (dots or asterisks).
  + **States:**
    - Same states as Text Input.
* **Dropdown Menu:**
  + **Usage:** For selecting from a list of options, such as roles or request types.
  + **Appearance:** White background, gray border, dropdown arrow.
  + **States:**
    - **Default:** Light gray border, placeholder text.
    - **Focus:** Blue border.
    - **Error:** Red border, with an error message below.
    - **Disabled:** Light gray background, dark gray border, no focus state.
* **Checkbox:**
  + **Usage:** For binary choices, like agreeing to terms or selecting multiple items.
  + **Appearance:** Square box, blue checkmark when selected.
  + **States:**
    - **Unchecked:** Empty box with gray border.
    - **Checked:** Box with blue checkmark.
    - **Disabled:** Light gray box, gray checkmark (if checked).
* **Radio Button:**
  + **Usage:** For selecting one option from a set.
  + **Appearance:** Circular button, filled when selected.
  + **States:**
    - **Unchecked:** Empty circle with gray border.
    - **Checked:** Filled circle with blue fill.
    - **Disabled:** Light gray circle, gray fill (if checked).
* **File Upload Field:**
  + **Usage:** For uploading files like documents or images.
  + **Appearance:** Text input with "Choose File" button, showing the file name after selection.
  + **States:**
    - **Default:** White background, gray border.
    - **Hover:** Blue border.
    - **Disabled:** Light gray background, gray border, no hover effect.

**3. Navigation Components**

* **Top Navigation Bar:**
  + **Usage:** For primary navigation between major sections of the application (e.g., Dashboard, Reports).
  + **Appearance:** Horizontal bar with links or icons, fixed at the top.
  + **States:**
    - **Default:** Light gray background, blue text for links.
    - **Hover:** Darker gray background, darker blue text.
    - **Active:** Blue background, white text for the active link.
* **Sidebar Menu:**
  + **Usage:** For secondary navigation within a section.
  + **Appearance:** Vertical sidebar with text links, collapsible.
  + **States:**
    - **Default:** Light gray background, blue text for links.
    - **Hover:** Darker gray background, darker blue text.
    - **Active:** Blue background, white text for the active link.
* **Breadcrumbs:**
  + **Usage:** To show the user’s location within the application.
  + **Appearance:** Horizontal text links separated by a greater-than sign ( > ).
  + **States:**
    - **Default:** Gray text for links, black text for the current page.
    - **Hover:** Blue text for links.
    - **Active:** No hover effect on the current page.

**4. Modals and Dialogs**

* **Modal Window:**
  + **Usage:** For important alerts, confirmations, or additional information that requires user interaction.
  + **Appearance:** Centered on the screen, slightly dimmed background, white content area with padding.
  + **States:**
    - **Default:** White background, gray border, shadow for elevation.
    - **Active:** Blue border or highlight to indicate focus.
* **Confirmation Dialog:**
  + **Usage:** To confirm actions like deletions or submissions.
  + **Appearance:** Smaller modal with action buttons (e.g., Confirm, Cancel).
  + **States:**
    - Same states as Modal Window, with primary action button (e.g., Confirm) highlighted.

**5. Tables and Lists**

* **Data Table:**
  + **Usage:** For displaying tabular data such as lists of purchase orders or users.
  + **Appearance:** Bordered table with alternating row colors for readability.
  + **States:**
    - **Default:** White background, gray borders, alternating light gray rows.
    - **Hover:** Highlighted row (e.g., light blue background).
    - **Selected:** Blue background for selected row, white text.
* **List View:**
  + **Usage:** For displaying items in a list format, such as notifications or tasks.
  + **Appearance:** Vertical list, with items separated by lines or padding.
  + **States:**
    - **Default:** White background, gray text.
    - **Hover:** Light blue background.
    - **Selected:** Blue background, white text.

**6. Notification Components**

* **Toast Notifications:**
  + **Usage:** For brief, temporary messages, like "Order submitted successfully."
  + **Appearance:** Small box that appears at the top or bottom of the screen, auto-dismisses after a few seconds.
  + **States:**
    - **Default:** Blue background, white text.
    - **Success:** Green background, white text.
    - **Error:** Red background, white text.
* **Alert Banner:**
  + **Usage:** For persistent alerts, like system messages or warnings.
  + **Appearance:** Full-width banner at the top or bottom of a section, with text and an optional close button.
  + **States:**
    - **Default:** Yellow background, black text.
    - **Success:** Green background, white text.
    - **Error:** Red background, white text.
    - **Info:** Blue background, white text.

**7. Icons**

* **Action Icons:**
  + **Usage:** For common actions like edit, delete, or settings.
  + **Appearance:** Simple, recognizable icons (e.g., pencil for edit, trash can for delete).
  + **States:**
    - **Default:** Gray icon.
    - **Hover:** Blue icon.
    - **Active:** Darker blue icon.
* **Status Icons:**
  + **Usage:** To indicate status, such as approved, pending, or error.
  + **Appearance:** Color-coded icons (e.g., checkmark for approved, exclamation mark for error).
  + **States:**
    - **Approved:** Green checkmark.
    - **Pending:** Yellow clock.
    - **Error:** Red exclamation mark.

**8. Typography**

* **Headings:**
  + **H1:** For page titles. Large, bold, primary color (e.g., blue).
  + **H2:** For section titles. Medium size, bold, dark gray.
  + **H3:** For sub-section titles. Slightly smaller, bold, dark gray.
* **Body Text:**
  + **Default:** Standard text size, dark gray, regular weight.
  + **Small Text:** Used for footnotes, captions, or less prominent information. Smaller size, gray color.
* **Links:**
  + **Default:** Blue text, underlined.
  + **Hover:** Darker blue text, underlined.
  + **Visited:** Purple text, underlined.
* **Style Guide:**
* **Consistency:** Use components consistently across the application to ensure a cohesive user experience. Buttons, form fields, and navigation elements should look and behave the same on every screen.
* **Accessibility:** Ensure that all components meet accessibility standards, such as sufficient color contrast, keyboard navigation, and screen reader compatibility.
* **Customization:** While the components should be standardized, allow for minor customizations (e.g., button colors or sizes) to fit specific use cases or sections of the application.
* **Documentation:** Include detailed documentation for each component, explaining when and how it should be used, along with any variations or states.

**5. Navigation Structure**

**1. Main Navigation Menu**

**Purpose:**  
The main navigation menu provides users with access to the primary sections of the application. It should be consistent across all screens and easily accessible.

**Structure:**

* **Dashboard:** The central hub where users can view an overview of their tasks, recent activities, and pending actions.
* **Create Purchase Order:** A direct link to the screen where users can create new purchase orders.
* **Pending Approvals:** Access to the list of purchase orders awaiting review and approval by the user.
* **Sessions:** Tools for creating and managing sessions that group multiple purchase orders for approval.
* **Reports:** Access to the reporting module, where users can generate and view reports.
* **Settings:** A section for user settings, including profile management, notification preferences, and other application configurations.
* **Help/Support:** Links to help documentation, FAQs, and contact options for support.

**Placement:**

* **Top Navigation Bar:** The main menu could be placed at the top of the screen, horizontally, with dropdowns for each main section if necessary.
* **Left Sidebar:** Alternatively, the main menu could be placed vertically on the left side of the screen, providing more space for nested items or additional links.

**Design Considerations:**

* **Icons and Labels:** Each menu item should be accompanied by an icon and a clear, concise label.
* **Active State:** The currently active section should be highlighted (e.g., with a different background color or underline).
* **Responsive Design:** Ensure that the menu is responsive, collapsing into a hamburger menu on smaller screens.

**2. Secondary Navigation**

**Purpose:**  
Secondary navigation options allow users to dive deeper into specific sections, such as accessing different sub-sections of a module.

**Structure:**

* **Dashboard Sub-Navigation:**
  + **Overview:** A summary of pending tasks and recent activity.
  + **Notifications:** A view of recent notifications and alerts.
  + **My Tasks:** A list of tasks assigned to the user.
* **Create Purchase Order Sub-Navigation:**
  + **New Order:** A direct link to start a new purchase order.
  + **Saved Drafts:** Access to any saved but not yet submitted purchase orders.
* **Pending Approvals Sub-Navigation:**
  + **All Approvals:** A comprehensive list of all pending approvals.
  + **Departmental Approvals:** Filtered view of approvals specific to the user's department.
  + **Urgent Approvals:** Approvals marked as high-priority.
* **Reports Sub-Navigation:**
  + **Generate Report:** Start the process of creating a new report.
  + **Saved Reports:** Access previously generated and saved reports.
  + **Report Templates:** Manage and select from report templates.

**Placement:**

* **Dropdowns:** If the main navigation is horizontal, secondary navigation items can be accessed via dropdown menus.
* **In-Page Navigation:** For screens like the dashboard, secondary navigation can be placed at the top of the content area, allowing users to switch between different views.

**Design Considerations:**

* **Visibility:** Secondary navigation should be easily visible and accessible, without overwhelming the user.
* **Clear Hierarchy:** Ensure that the relationship between primary and secondary navigation items is clear.

**3. Breadcrumbs**

**Purpose:**  
Breadcrumbs provide a trail of links that shows the user’s location within the application, helping them navigate back to previous pages or sections.

**Structure:**

* **Home / Dashboard / Section / Sub-section:** A simple breadcrumb structure showing the path from the home or dashboard to the current page.

**Placement:**

* **Top of Content Area:** Breadcrumbs are typically placed at the top of the content area, just below the main navigation bar.

**Design Considerations:**

* **Separator:** Use a simple separator (e.g., “>” or “/”) between breadcrumb items.
* **Interactive:** Each part of the breadcrumb (except the current page) should be clickable, allowing users to navigate back to previous sections easily.

**4. Quick Links and Shortcuts**

**Purpose:**  
Quick links and shortcuts provide users with fast access to commonly used features or sections.

**Structure:**

* **Dashboard Quick Links:**
  + **Create New Purchase Order:** A button or link to start a new purchase order directly from the dashboard.
  + **View Pending Approvals:** Quick access to the list of pending approvals.
  + **Generate Report:** A link to start the report generation process.
* **Footer Shortcuts:**
  + **Help:** A quick link to the help center or support.
  + **Settings:** Direct access to user settings and preferences.
  + **Log Out:** A logout button for quick exit.

**Placement:**

* **Dashboard:** Quick links can be placed prominently on the dashboard, either at the top or in a sidebar.
* **Footer:** Shortcuts can be placed in the footer of the application, accessible from any page.

**Design Considerations:**

* **Prominence:** Ensure that quick links and shortcuts are prominent but do not clutter the UI.
* **Icons:** Use icons alongside text labels to make these links easily recognizable.

**5. Search Bar Navigation**

**Purpose:**  
A global search bar allows users to quickly find specific content, such as purchase orders, reports, or users, without navigating through multiple menus.

**Structure:**

* **Search Scope:** Allow users to specify what they are searching for (e.g., Purchase Orders, Reports, Users).
* **Search Results Page:** A dedicated page that displays search results in a list or table format, with options to filter and sort the results.

**Placement:**

* **Top Navigation Bar:** The search bar is often placed prominently in the top navigation bar, making it accessible from any page.
* **Dashboard:** An additional search bar can be included on the dashboard for quick access.

**Design Considerations:**

* **Auto-suggestions:** Implement auto-suggestions or autocomplete to help users find what they are looking for faster.
* **Advanced Search:** Offer an advanced search option with additional filters for more precise queries.

**6. Responsive Design Considerations**

**Purpose:**  
To ensure that the navigation structure is accessible and functional across different devices and screen sizes.

**Structure:**

* **Mobile Navigation:**
  + **Hamburger Menu:** Collapse the main navigation into a hamburger menu for smaller screens.
  + **Bottom Navigation:** For mobile, consider using a bottom navigation bar for easier access on touch screens.
* **Tablet Navigation:**
  + **Combined Sidebar/Menu:** On tablets, a combined sidebar that collapses into icons can be useful to save space while maintaining accessibility.

**Design Considerations:**

* **Touch Targets:** Ensure that all navigation elements have sufficiently large touch targets to accommodate finger taps.
* **Simplified Menus:** On smaller screens, consider simplifying the navigation structure by reducing the number of visible items.

**6. Interaction Design**

**1. Button Interactions**

**1.1. Primary Button (e.g., "Submit")**

* **Default State:**
  + Appearance: Solid blue background, white text.
  + Behavior: Button is enabled and clickable.
* **Hover State:**
  + Appearance: Darker blue background, white text.
  + Behavior: Button changes color to indicate it is interactive.
* **Active State:**
  + Appearance: Even darker blue, slight inset effect to simulate pressing.
  + Behavior: Button gives feedback that it has been clicked, triggering the associated action (e.g., form submission).
* **Disabled State:**
  + Appearance: Light gray background, gray text.
  + Behavior: Button is non-clickable, indicating that the action is not currently available (e.g., due to incomplete form fields).

**1.2. Secondary Button (e.g., "Cancel")**

* **Default State:**
  + Appearance: Transparent background, blue border, blue text.
  + Behavior: Button is enabled and clickable.
* **Hover State:**
  + Appearance: Light blue background, blue text.
  + Behavior: Button highlights to indicate interactivity.
* **Active State:**
  + Appearance: Darker blue border, inset effect.
  + Behavior: Button is clicked, triggering the cancel action.
* **Disabled State:**
  + Appearance: Transparent background, gray border, gray text.
  + Behavior: Button is non-clickable, action not available.

**1.3. Icon Button (e.g., "Search")**

* **Default State:**
  + Appearance: Gray icon, no background.
  + Behavior: Icon is clickable and performs the associated action (e.g., opens search).
* **Hover State:**
  + Appearance: Blue icon.
  + Behavior: Icon color changes to indicate it is interactive.
* **Active State:**
  + Appearance: Darker blue icon.
  + Behavior: Icon is clicked, and the associated action (e.g., opening a search bar) is triggered.

**2. Form Element Interactions**

**2.1. Text Input Fields**

* **Default State:**
  + Appearance: White background, light gray border, dark gray text.
  + Behavior: Input is editable.
* **Focus State:**
  + Appearance: Blue border, slight shadow.
  + Behavior: Field is active, ready to receive user input.
* **Error State:**
  + Appearance: Red border, error message displayed below the field.
  + Behavior: Indicates invalid or missing input.
* **Disabled State:**
  + Appearance: Light gray background, dark gray border, no input allowed.
  + Behavior: Field is not editable, possibly due to form or process conditions.

**2.2. Dropdown Menus**

* **Default State:**
  + Appearance: White background, gray border, dropdown arrow.
  + Behavior: Shows placeholder text or selected value.
* **Open State:**
  + Appearance: Dropdown list expands, blue border.
  + Behavior: User can scroll and select an option.
* **Error State:**
  + Appearance: Red border, error message displayed below.
  + Behavior: Indicates an invalid or required selection.
* **Disabled State:**
  + Appearance: Light gray background, gray border, dropdown is non-interactive.
  + Behavior: Dropdown cannot be opened or changed.

**2.3. File Upload**

* **Default State:**
  + Appearance: Text input with "Choose File" button.
  + Behavior: Users can click the button to open the file dialog.
* **File Selected:**
  + Appearance: Displays the file name in the input area.
  + Behavior: User sees the file selected, with an option to remove or replace the file.
* **Error State:**
  + Appearance: Red border or text, error message displayed.
  + Behavior: Indicates an issue with the file (e.g., wrong format, size too large).

**3. Navigation Interactions**

**3.1. Main Navigation Menu**

* **Default State:**
  + Appearance: Horizontal or vertical bar with icons and labels.
  + Behavior: Menu items are visible and clickable.
* **Hover State:**
  + Appearance: Background changes color (e.g., light blue), text remains the same.
  + Behavior: Indicates that the menu item is interactive.
* **Active State:**
  + Appearance: Selected menu item is highlighted (e.g., blue background, white text).
  + Behavior: Indicates the current section the user is viewing.
* **Dropdown Interaction (if applicable):**
  + Behavior: When a user hovers or clicks on a menu item, a dropdown expands, showing sub-menu items.

**3.2. Sidebar Navigation**

* **Default State:**
  + Appearance: Vertical bar with menu items stacked.
  + Behavior: Menu items are visible, and the current section is highlighted.
* **Collapse/Expand Interaction:**
  + Behavior: Sidebar can be collapsed into icons only, or expanded to show full labels. Clicking an icon in collapsed mode expands the sidebar.

**3.3. Breadcrumb Navigation**

* **Default State:**
  + Appearance: Horizontal text links separated by a “>” or “/”.
  + Behavior: Links show the user’s current path in the application.
* **Hover State:**
  + Appearance: Link text changes color (e.g., from gray to blue).
  + Behavior: Indicates that the breadcrumb is clickable.
* **Active State:**
  + Appearance: The current page is bolded or highlighted, with no hover effect.
  + Behavior: Shows the user’s current location.

**4. Modals and Dialogs**

**4.1. Modal Windows**

* **Open Interaction:**
  + Behavior: When triggered (e.g., by a button click), the modal smoothly fades in, with the background dimmed.
* **Focus Management:**
  + Behavior: Focus should be automatically set to the modal content or the first actionable item within the modal.
* **Close Interaction:**
  + Behavior: Users can close the modal by clicking a close icon (typically a “X” in the top-right corner), clicking outside the modal area, or pressing the “Esc” key. The modal smoothly fades out, returning focus to the previous active element.

**4.2. Confirmation Dialogs**

* **Open Interaction:**
  + Behavior: Dialog opens in the center of the screen, with a message and action buttons (e.g., Confirm, Cancel).
* **Confirm/Cancel Interaction:**
  + Behavior: Clicking “Confirm” triggers the action and closes the dialog. Clicking “Cancel” closes the dialog without action.
* **Error Handling:**
  + Behavior: If the confirmation fails (e.g., due to a system error), an error message is displayed, and the user remains in the dialog.

**5. Feedback Mechanisms**

**5.1. Form Validation Feedback**

* **Real-time Validation:**
  + Behavior: As the user types or selects an option, the system provides immediate feedback (e.g., error message for an invalid email format).
* **Submit-time Validation:**
  + Behavior: When the user submits the form, the system checks all fields and provides feedback on any missing or incorrect information.
* **Success Feedback:**
  + Behavior: Upon successful submission, a confirmation message is displayed (e.g., “Purchase order submitted successfully”), and the form resets or redirects the user.

**5.2. Toast Notifications**

* **Trigger:**
  + Behavior: Notifications are triggered by specific events, such as successful actions (e.g., “Order saved successfully”) or errors (e.g., “Failed to upload file”).
* **Appearance:**
  + Behavior: Toast appears at the top-right or bottom-right of the screen and auto-dismisses after a few seconds. Users can manually dismiss by clicking on the notification.
* **Error Handling:**
  + Behavior: Error notifications remain visible longer and may include a “Retry” button or additional instructions.

**5.3. Loading Indicators**

* **Page Load:**
  + Behavior: A full-screen spinner or progress bar appears while the page content is loading.
* **Form/Action Submission:**
  + Behavior: A smaller inline spinner appears on the button or near the field where the action is taking place, indicating processing.
* **Asynchronous Data Fetch:**
  + Behavior: A loading spinner or “Loading…” message appears within the data area while content is being fetched.

**6. Error Handling and Alerts**

**6.1. Inline Error Messages**

* **Trigger:**
  + Behavior: Inline error messages are displayed near the relevant field when an error is detected (e.g., invalid input).
* **Appearance:**
  + Behavior: The field’s border turns red, and a small error message is displayed below the field in red text.
* **Resolution:**
  + Behavior: The error message disappears once the input is corrected, and the field returns to its normal state.

**6.2. Alert Banners**

* **Trigger:**
  + Behavior: Alert banners appear at the top of the page in response to significant system events (e.g., server maintenance, critical error).
* **Appearance:**
  + Behavior: The banner has a strong background color (e.g., red for errors, yellow for warnings) and white or black text.
* **Dismissal:**
  + Behavior: Users can dismiss the banner by clicking an “X” icon, if applicable.

**6.3. Error Pages**

* **Trigger:**
  + Behavior: Error pages are shown when a user encounters a critical issue, such as a 404 (Page Not Found) or 500 (Server Error).
* **Appearance:**
  + Behavior: The error page displays a friendly error message, an error code, and options to return to the previous page or go to the home page.
* **Navigation:**
  + Behavior: Include links to helpful resources (e.g., support, FAQs) on the error page.

**7. Responsive Interactions**

**7.1. Mobile Navigation**

* **Hamburger Menu:**
  + Behavior: On small screens, the main navigation collapses into a hamburger menu, which expands into a full-screen overlay when tapped.
* **Touch-Friendly Elements:**
  + Behavior: Buttons and interactive elements are larger to accommodate touch inputs, with more spacing between elements to prevent accidental taps.

**7.2. Adaptive Layouts**

* **Form Fields:**
  + Behavior: Form fields stack vertically on small screens, and labels move above inputs to save space.
* **Modals:**
  + Behavior: Modals expand to fill the screen or take up a larger portion of the screen on mobile devices, with a clear “Close” button.

**7. Responsive Design Considerations**

**1. Screen Resolution Adaptation**

**1.1. High Resolution (e.g., 4K monitors)**

* **Layout Scaling:** Ensure that the application scales appropriately on high-resolution monitors. Elements should not appear too small; consider using vector graphics and scalable UI components.
* **Font Scaling:** Fonts should be sharp and readable at high resolutions. Support for Windows' DPI scaling settings is important.

**1.2. Standard Resolution (e.g., 1080p)**

* **Default Layout:** Design the application with 1080p as a baseline resolution, ensuring all elements are appropriately sized and spaced.
* **Window Resizing:** The application should allow users to resize windows, with content adapting to fill the available space without breaking the layout.

**1.3. Low Resolution (e.g., older monitors)**

* **Compact Layout:** Ensure the application is usable on lower resolutions, like 1366x768, by providing a compact layout that avoids horizontal scrolling and keeps critical information accessible.
* **Scrollbars:** Vertical scrollbars should appear where content overflows vertically, but horizontal scrolling should generally be avoided.

**2. Window Resizing and Adaptive Layouts**

* **Resizable Windows:**
  + Allow users to resize application windows. The layout should adapt dynamically, such as moving elements or resizing panels to fit the available space.
* **Minimizing and Maximizing:**
  + The application should handle window minimizing and maximizing gracefully. Upon maximizing, the UI should scale to fill the screen without stretching or distorting elements.
* **Adaptive Panels:**
  + Consider using collapsible or expandable panels. For example, a sidebar might collapse into icons when the window is too narrow, allowing more space for the main content.

**3. Multi-Monitor Support**

* **Multi-Monitor Layouts:**
  + Ensure that the application functions correctly when moved between monitors of different resolutions and scaling settings.
* **Floating Windows:**
  + If the application uses floating windows (e.g., tool palettes or secondary dialogs), ensure they remain attached to the main window or the monitor where they were opened.

**4. Font and UI Scaling**

* **DPI Awareness:**
  + Ensure that the application is DPI-aware, meaning it should respect the user’s display settings in Windows. This includes scaling fonts, icons, and other UI elements accordingly.
* **Accessibility Settings:**
  + Consider providing options within the application to increase font size or switch to a high-contrast theme for users with visual impairments.

**5. Touch Interaction (if applicable)**

* **Touchscreen Monitors:**
  + If the application might be used on touchscreen monitors, consider making buttons and other interactive elements large enough to be easily tappable. This is especially important for all-in-one PCs or Windows tablets used in desktop mode.
* **Pen Input:**
  + If relevant, ensure that the application supports pen input, such as for drawing or signing documents.

**6. Responsive Design Elements within the Desktop Context**

* **Adaptive Layouts for Split-Screen:**
  + Consider users who might use the application in split-screen mode (e.g., with another application side-by-side). Ensure that key information remains visible and accessible when the window is not full-screen.
* **Popup and Modal Dialogs:**
  + Design popups and modal dialogs to be appropriately sized and centered, with consideration for smaller screen sizes or constrained window sizes.

**8. Prototyping**

**1. User Flow: Creating and Submitting a Purchase Order**

**Persona:** Luis García – Purchasing Employee

**Objective:** Create and submit a purchase order for approval.

**Steps:**

1. **Login to the Application**
   * **Action:** Luis opens the CEA application and enters his username and password on the login screen.
   * **Expected Result:** Luis is successfully logged in and redirected to the dashboard.
2. **Navigate to the Purchase Order Creation Screen**
   * **Action:** From the dashboard, Luis clicks on the "Create Purchase Order" button in the main navigation or via a quick link.
   * **Expected Result:** The Purchase Order Creation screen is displayed.
3. **Fill in Purchase Order Details**
   * **Action:** Luis fills out all mandatory fields, including "Order Title," "Amount," "Vendor," "Contract Number," etc.
   * **Expected Result:** The fields are populated with the entered data, and validation messages appear if any required fields are missing.
4. **Upload Vendor Comparison Document**
   * **Action:** Luis clicks the "Upload Document" button and selects a comparison table image from his computer.
   * **Expected Result:** The file is uploaded successfully, and the file name is displayed in the upload field.
5. **Review and Submit the Purchase Order**
   * **Action:** Luis reviews the entered information and clicks the "Submit Purchase Order" button.
   * **Expected Result:** The system validates the form, submits the purchase order, and displays a confirmation message. Luis is redirected to the dashboard, where the submitted order appears in the "Pending Orders" section.
6. **Track Order Status**
   * **Action:** Luis checks the status of the submitted order on the dashboard.
   * **Expected Result:** The order status is displayed as "Pending Approval" with a link to view detailed information.

**2. User Flow: Reviewing and Approving a Purchase Order**

**Persona:** Marta Rivera – Purchasing Supervisor

**Objective:** Review and approve a submitted purchase order.

**Steps:**

1. **Login to the Application**
   * **Action:** Marta opens the CEA application and enters her credentials.
   * **Expected Result:** Marta is successfully logged in and redirected to her personalized dashboard.
2. **Access the Pending Approvals Screen**
   * **Action:** From the dashboard, Marta clicks on the "Pending Approvals" link in the main navigation.
   * **Expected Result:** The Pending Approvals screen is displayed, listing all purchase orders awaiting her review.
3. **Select a Purchase Order for Review**
   * **Action:** Marta clicks on a specific purchase order from the list to view its details.
   * **Expected Result:** The Purchase Order Details screen is displayed, showing all relevant information and the uploaded vendor comparison document.
4. **Review the Purchase Order Details**
   * **Action:** Marta carefully reviews the order details and checks the attached document.
   * **Expected Result:** The details are accurate and clear, with the document preview or download option available.
5. **Approve, Modify, or Reject the Order**
   * **Action:** Marta decides to approve the order and clicks the "Approve" button.
   * **Expected Result:** A confirmation dialog appears, Marta confirms her decision, and the order status updates to "Approved." If she chooses to modify or reject, the system prompts for additional information or feedback.
6. **Monitor Approval Progress**
   * **Action:** Marta returns to the Pending Approvals screen to see the updated status of the approved order.
   * **Expected Result:** The order is no longer listed under pending approvals, and its status is updated accordingly.

**3. User Flow: Generating and Viewing a Report**

**Persona:** Carlos López – Purchasing Manager

**Objective:** Generate a report summarizing the department’s purchase orders for the last quarter.

**Steps:**

1. **Login to the Application**
   * **Action:** Carlos logs into the CEA application with his username and password.
   * **Expected Result:** Carlos is successfully logged in and taken to the dashboard.
2. **Navigate to the Reports Section**
   * **Action:** Carlos clicks on the "Reports" link in the main navigation bar.
   * **Expected Result:** The Reports screen is displayed, showing various reporting options.
3. **Select Report Parameters**
   * **Action:** Carlos selects the report type (e.g., "Quarterly Summary") and sets the date range for the last quarter.
   * **Expected Result:** The parameters are set, and the system is ready to generate the report.
4. **Generate the Report**
   * **Action:** Carlos clicks the "Generate Report" button.
   * **Expected Result:** The system processes the request and displays the generated report in the report viewer.
5. **Review and Export the Report**
   * **Action:** Carlos reviews the report and decides to export it as a PDF.
   * **Expected Result:** The report is exported successfully and saved to Carlos's computer.

**4. User Flow: Managing a Session**

**Persona:** Ana Martínez – Director of Purchasing

**Objective:** Create a session to group multiple purchase orders for final approval by the general direction.

**Steps:**

1. **Login to the Application**
   * **Action:** Ana logs into the CEA application with her credentials.
   * **Expected Result:** Ana is successfully logged in and directed to her dashboard.
2. **Navigate to the Session Management Screen**
   * **Action:** Ana clicks on the "Sessions" link in the main navigation.
   * **Expected Result:** The Session Management screen is displayed, showing existing sessions and an option to create a new one.
3. **Create a New Session**
   * **Action:** Ana clicks the "Create New Session" button, enters a session title and description, and selects purchase orders to include in the session.
   * **Expected Result:** The session is created, and the selected orders are grouped under this session.
4. **Review and Submit the Session**
   * **Action:** Ana reviews the session details and clicks the "Submit for Final Approval" button.
   * **Expected Result:** The session is submitted, and Ana receives a confirmation that it has been forwarded to the general direction for approval.
5. **Monitor Session Status**
   * **Action:** Ana returns to the Session Management screen to monitor the approval progress of the submitted session.
   * **Expected Result:** The session status is updated in real-time as approvals are received.

**Preparing for User Testing**

To prepare for user testing, these user flows will serve as the basis for the test scenarios. Here’s how to document and conduct user testing based on the simulated flows:

**User Testing Preparation**

1. **Define Test Scenarios**
   * Use the simulated user flows to create specific test scenarios. Each scenario should describe the task the user needs to complete and the expected outcomes.
2. **Prepare Test Scripts**
   * Write test scripts that guide users through the test scenarios. Include instructions, tasks, and prompts to capture user feedback.
3. **Set Up the Prototype**
   * Ensure the prototype is functional and includes all the screens and interactions outlined in the user flows. Tools like Figma, Adobe XD, or InVision can be used to create interactive prototypes.
4. **Recruit Test Participants**
   * Select participants that match the user personas (e.g., actual employees who will use the application).
5. **Conduct the Tests**
   * Have participants complete the test scenarios while observing their interactions, collecting feedback, and noting any issues or points of confusion.
6. **Analyze Results**
   * Review the test results to identify patterns, usability issues, and areas for improvement. Summarize findings and suggest changes to the design.
7. **Iterate on the Design**
   * Based on user testing feedback, refine the design and interactions. Update the prototype and prepare for further testing if necessary.

**9. Final Design Specification**

**1. Introduction**

**1.1. Overview**

* **Project Name:** CEA Application
* **Purpose:** The CEA application is designed to streamline the purchase order process for purchases exceeding 1 million Mexican pesos. It includes features for creating, approving, and managing purchase orders, as well as generating reports and managing sessions for final approval.
* **Target Users:** Purchasing employees, supervisors, managers, and directors within the organization.

**1.2. Document Purpose**

* This document provides detailed design specifications for the CEA application, including UI components, interaction design, navigation structure, responsive design considerations, and final prototype details. It serves as the blueprint for the development and implementation of the application.

**1.3. Scope**

* The document covers the complete design of the CEA application, including user interface (UI) components, user interactions, screen layouts, navigation flows, and responsive behavior. It also includes specifications for final prototype and user testing feedback.

**2. UI Components Library**

**2.1. Buttons**

* **Primary Button:**
  + **Style:** Solid blue background, white text, rounded corners.
  + **States:** Default, Hover, Active, Disabled.
* **Secondary Button:**
  + **Style:** Transparent background, blue border, blue text.
  + **States:** Default, Hover, Active, Disabled.
* **Icon Button:**
  + **Style:** Icon-only, gray by default, blue on hover.
  + **Usage:** For actions like search, settings.

**2.2. Form Elements**

* **Text Input:**
  + **Style:** White background, gray border, dark gray text.
  + **States:** Default, Focus, Error, Disabled.
* **Dropdown Menu:**
  + **Style:** White background, gray border, dropdown arrow.
  + **States:** Default, Open, Error, Disabled.
* **Checkbox:**
  + **Style:** Square box, blue checkmark when selected.
  + **States:** Unchecked, Checked, Disabled.
* **Radio Button:**
  + **Style:** Circular button, blue fill when selected.
  + **States:** Unchecked, Checked, Disabled.
* **File Upload Field:**
  + **Style:** Text input with "Choose File" button.
  + **States:** Default, Hover, Disabled.

**2.3. Navigation Components**

* **Main Navigation Bar:**
  + **Style:** Horizontal bar with icons and text labels.
  + **Placement:** Top of the screen, responsive design collapses to a hamburger menu on smaller screens.
* **Sidebar Menu:**
  + **Style:** Vertical sidebar, collapsible, with icons and text.
  + **Placement:** Left side of the screen.
* **Breadcrumbs:**
  + **Style:** Horizontal text links, separated by ">", showing the user's location.
  + **Placement:** Below the main navigation bar.

**2.4. Modals and Dialogs**

* **Modal Window:**
  + **Style:** Centered on the screen, white background, shadow, and close button.
  + **Behavior:** Appears with a fade-in effect, closes on clicking "X" or outside the modal.
* **Confirmation Dialog:**
  + **Style:** Smaller modal, includes action buttons (e.g., Confirm, Cancel).
  + **Behavior:** Triggered by actions requiring confirmation, e.g., deleting an item.

**2.5. Tables and Lists**

* **Data Table:**
  + **Style:** Bordered table, alternating row colors for readability.
  + **States:** Default, Hover, Selected.
* **List View:**
  + **Style:** Vertical list with separators between items.
  + **States:** Default, Hover, Selected.

**2.6. Typography**

* **Headings:**
  + **H1:** Large, bold, primary color for page titles.
  + **H2:** Medium size, bold for section titles.
  + **H3:** Smaller, bold for sub-section titles.
* **Body Text:**
  + **Default:** Standard text size, dark gray.
  + **Small Text:** For captions, footnotes.
* **Links:**
  + **Style:** Blue text, underlined.
  + **States:** Default, Hover, Visited.

**3. Interaction Design Specifications**

**3.1. Button Interactions**

* **Primary Button:**
  + **Default:** Solid blue with white text.
  + **Hover:** Darker blue.
  + **Active:** Inset effect, indicates action is being performed.
* **Secondary Button:**
  + **Default:** Transparent with blue border.
  + **Hover:** Light blue background.

**3.2. Form Elements**

* **Text Input Fields:**
  + **Focus State:** Blue border appears when the field is active.
  + **Error State:** Red border with an error message below the field.
* **Dropdown Menus:**
  + **Open State:** Dropdown expands, displaying options.
  + **Error State:** Red border indicates selection is required or invalid.

**3.3. Navigation**

* **Main Navigation Bar:**
  + **Hover State:** Background color changes to indicate interactivity.
  + **Active State:** The current section is highlighted with a different background.
* **Sidebar Navigation:**
  + **Collapse/Expand:** The sidebar can be collapsed to icons only, expanding on hover or click.

**3.4. Feedback Mechanisms**

* **Form Validation:**
  + **Real-time Validation:** Immediate feedback as the user types or selects.
  + **Submit-time Validation:** Errors are displayed upon form submission if any fields are incomplete or incorrect.
* **Toast Notifications:**
  + **Appearance:** Top-right of the screen, fades out after a few seconds.
  + **Types:** Success, Error, Information.
* **Loading Indicators:**
  + **Behavior:** Spinner or progress bar appears while content is loading.

**3.5. Error Handling**

* **Inline Error Messages:**
  + **Placement:** Displayed near the field with the issue.
  + **Behavior:** Red text with a short explanation of the error.
* **Alert Banners:**
  + **Appearance:** Full-width banner at the top of the screen, with dismissible options.
  + **Types:** Warning, Error, Success.

**4. Responsive Design Considerations**

**4.1. Screen Resolution Adaptation**

* **High Resolution (4K Monitors):**
  + Ensure UI components scale appropriately, maintaining readability and usability.
* **Standard Resolution (1080p):**
  + Design with 1080p as the baseline, ensuring that content is well-spaced and accessible.
* **Low Resolution:**
  + Provide a compact layout that is still functional, with necessary scrollbars where content overflows.

**4.2. Window Resizing and Adaptive Layouts**

* **Resizable Windows:**
  + The layout should adapt dynamically to window resizing, maintaining usability.
* **Minimizing and Maximizing:**
  + Handle these actions gracefully, ensuring the UI scales without distortion.
* **Adaptive Panels:**
  + Use collapsible or expandable panels to manage space effectively.

**4.3. Multi-Monitor Support**

* **Multi-Monitor Layouts:**
  + Ensure the application works well when moved between monitors with different resolutions or scaling settings.
* **Floating Windows:**
  + Ensure floating windows (e.g., tool palettes) remain attached to the main window or monitor where they were opened.

**4.4. Font and UI Scaling**

* **DPI Awareness:**
  + Ensure the application is DPI-aware, scaling fonts and UI elements appropriately for the user's display settings.
* **Accessibility Settings:**
  + Provide options to increase font size or switch to a high-contrast theme.

**5. Navigation Structure**

**5.1. Main Navigation**

* **Structure:**
  + Links to key sections: Dashboard, Create Purchase Order, Pending Approvals, Sessions, Reports, Settings, Help/Support.
* **Placement:**
  + Horizontal bar at the top or vertical sidebar on the left.
* **Responsive Behavior:**
  + Collapses into a hamburger menu on smaller screens.

**5.2. Secondary Navigation**

* **Structure:**
  + Sub-navigation for each main section, providing access to detailed views or additional features.
* **Placement:**
  + Dropdowns or in-page tabs, depending on the screen layout.
* **Responsive Behavior:**
  + Collapses or becomes part of the hamburger menu on smaller screens.

**5.3. Breadcrumbs**

* **Structure:**
  + Shows the user’s location within the application (e.g., Home > Dashboard > Pending Approvals).
* **Placement:**
  + Below the main navigation bar, always visible.
* **Responsive Behavior:**
  + Breadcrumbs shorten or collapse on smaller screens.

**6. Prototype and User Testing**

**6.1. Prototype Details**

* **Tool Used:**
  + (Specify the tool, e.g., Figma, Adobe XD).
* **Interactive Features:**
  + The prototype includes clickable elements, modal pop-ups, and form validations.
* **Screens Included:**
  + All key screens (Login, Dashboard, Purchase Order Creation, Pending Approvals, Reports, etc.).

**6.2. User Testing Feedback**

* **Summary of Testing:**
  + (Summarize the results of user testing, including any issues identified and resolved).
* **Design Iterations:**
  + (Detail any design changes made based on testing feedback).

**6.3. Final Prototype Link**

* **Access:**
  + (Provide a link to the final prototype for stakeholders and developers).

**7. Final Design Review**

**7.1. Design Approval**

* **Stakeholders:**
  + (List the stakeholders who approved the final design).
* **Approval Date:**
  + (Specify the date of approval).

**7.2. Handoff to Development**

* **Design Handoff Tool:**
  + (Specify the tool used for handoff, e.g., Zeplin, Figma Inspect).
* **Handoff Details:**
  + All design assets, including UI components, icons, images, and fonts, are included with the necessary specifications for developers.

**7.3. Ongoing Collaboration**

* **Communication Channel:**
  + (Specify how designers and developers will communicate during implementation, e.g., Slack, Microsoft Teams).
* **Review Process:**
  + (Outline the process for reviewing implemented designs against the specifications).

**8. Appendices**

**8.1. Glossary**

* **Terms:**
  + Provide definitions for any technical terms or abbreviations used in the documentation.

**8.2. Style Guide**

* **Brand Colors:**
  + Specify the exact colors used in the application, including HEX, RGB, and CMYK codes.
* **Typography:**
  + Include font names, sizes, and weights.
* **Iconography:**
  + Provide a list of icons used, including style guidelines.

**8.3. Resources**

* **Links to Design Files:**
  + Provide links to any relevant design files, such as the UI components library, wireframes, and mockups.

**8.4. Contact Information**

* **Design Team:**
  + List the members of the design team, along with their roles and contact information.

**10. Ongoing Maintenance and Updates**

* **Version Control:**
  + Keep track of changes in the UI/UX design over time.
* **Continuous Improvement:**
  + Regularly gather user feedback post-launch and iterate on the design as necessary.