

Intranet Portal at Zewail City of Science and Technology

Software Requirements Document

1. Document Control

* Document Title: Intranet Portal Software Requirements
* Date: [Dec 1st, 2024]
* Version: 1.0

Part 1: Functional Requirements

1. User Interface Design

Goal: Provide a scalable and streamlined interface that enhances user experience and supports future feature expansion.

* Requirements:
  1. Develop a modular design framework that minimizes visual clutter.
  2. Utilize responsive web design to ensure usability across devices (desktop, tablet, and mobile browsers).
  3. Incorporate user feedback mechanisms for iterative improvement.
  4. Build a flexible layout allowing integration of additional modules without major redesigns.

2. Mobile Application Development

Goal: Build Android and iOS applications using Flutter for seamless access to the intranet portal.

* Requirements:
  1. Design and develop a cross-platform application ensuring consistent UI/UX on both platforms.
  2. Integrate core functionalities such as notifications, service requests, attendance, and HR requests.
  3. Implement offline capabilities for key features (e.g., access to announcements).
  4. Ensure performance optimization for low-resource devices.

3. iTop Service Management Integration  
iTop software (an open-source IT service management tool) <https://www.combodo.com/itop-193>  
Goal: Integrate iTop as the primary tool for IT ticketing and service management.

* Requirements:
  1. Configure iTop to automatically log all IT service requests raised via the portal.
  2. Design user-friendly forms for IT requests that map directly to iTop ticket categories.
  3. Provide role-based dashboards in iTop for IT team members and stakeholders.
  4. Enable email and portal notifications for ticket status updates.

Integrating iTop, an open-source IT Service Management (ITSM) and Configuration Management Database (CMDB) system, with Microsoft Power Apps and Power Apps Portals can enhance your IT operations by enabling custom applications that interact seamlessly with your IT management data. Here's a step-by-step guide to achieve this integration:

**1. Understand iTop's REST/JSON API:**

iTop provides a REST/JSON API that allows external applications to interact with its data. This API supports operations such as retrieving and updating records, which are essential for integration. Detailed documentation and examples are available to help you understand the API's capabilities.

[Postman](https://www.postman.com/combodo/combodo-s-public-workspace/documentation/svyrnfy/itop-rest-json?utm_source=chatgpt.com)

**2. Create a Custom Connector in Power Apps:**

To enable Power Apps to communicate with iTop's API, you'll need to create a custom connector:

* **Define the Connector:**
  + Navigate to the Power Apps portal and select "Custom Connectors."
  + Choose to create a new connector from blank.
  + Provide a name and description for your connector.
* **Set Up the API Connection:**
  + In the connector settings, specify the base URL of your iTop instance.
  + Configure authentication. iTop's API typically uses Basic Authentication, so you'll need to set this up accordingly.
* **Define Actions:**
  + For each API operation you wish to perform (e.g., retrieving incidents, updating tickets), define an action in the connector.
  + Specify the request method (GET, POST, etc.), URL, headers, and parameters based on iTop's API documentation.
* **Test the Connector:**
  + Use the built-in testing tools to ensure that each action communicates correctly with iTop.

Microsoft provides comprehensive guidance on creating custom connectors, which can be invaluable during this process.

[Microsoft Learn](https://learn.microsoft.com/en-us/connectors/custom-connectors/?utm_source=chatgpt.com)

**3. Integrate the Connector into Power Apps and Power Apps Portals:**

* **In Power Apps:**
  + Add the custom connector to your app as a data source.
  + Use the connector's actions within your app to perform operations like displaying iTop data or creating new records.
* **In Power Apps Portals:**
  + Utilize the custom connector to expose iTop data to portal users.
  + Ensure that appropriate permissions and security measures are in place to control data access.

**4. Implement Security Measures:**

Given that iTop's API uses Basic Authentication, it's crucial to handle credentials securely:

* **Secure Storage:** Store API credentials securely, avoiding hardcoding them into applications.
* **HTTPS:** Ensure that all API communications occur over HTTPS to encrypt data in transit.
* **Access Control:** Define and enforce access controls to restrict who can perform specific actions via the API.

**5. Maintain and Update the Integration:**

Regularly monitor the integration to ensure it functions correctly:

* **API Changes:** Stay informed about updates to iTop's API that might affect your connector.
* **Connector Updates:** Update the custom connector as needed to accommodate new requirements or changes in the API.
* **Error Handling:** Implement robust error handling to manage issues like network failures or unexpected API responses.

By following these steps, you can create a seamless integration between iTop and Microsoft Power Apps, enhancing your IT management capabilities and providing users with intuitive access to essential data.

4. System Integration Planning

Goal: Facilitate secure and efficient integration of internal systems.

* Requirements:
  1. Map current and planned integrations for the ERP system, attendance tracking, and HR systems.
  2. Perform security assessments to identify and mitigate potential vulnerabilities.
  3. Utilize APIs or middleware for secure data exchange between integrated systems.
  4. Document integration workflows and provide maintenance guidelines.

5. Transition to a Paperless Organization

Goal: Replace paper forms with digital forms integrated with the ERP system.

* Requirements:
  1. Analyze existing paper-based workflows to identify conversion requirements.
  2. Develop digital forms using a portal-based framework, ensuring compatibility with the ERP system.
  3. Provide digital signatures and approval workflows for form submissions.
  4. Train employees on using the new system to ensure a smooth transition.

6. Attendance Tracking System

Goal: Develop a portal-based attendance tracking system.

* Requirements:
  1. Allow employees to view their attendance records through a secure portal interface.
  2. Enable managers and HR to access hierarchical views of attendance data.
  3. Integrate with biometric or card-based attendance devices (if applicable).
  4. Provide real-time analytics and reports for decision-making.

7. HR Request System

Goal: Digitize HR-related requests, such as leave applications, hiring, and attendance issues.

* Requirements:
  1. Collaborate with HR to define and categorize request types.
  2. Design a portal-based submission and tracking system for HR requests.
  3. Integrate approval workflows based on hierarchical roles.
  4. Enable notifications for request status updates.

8. Information Portal

Goal: Centralize process documents and announcements for easy access.

* Requirements:
  1. Develop a searchable, user-friendly portal for storing and accessing documents.
  2. Implement version control and permissions for managing sensitive information.
  3. Update the portal regularly with relevant announcements and procedural updates.
  4. Allow tagging and categorization for intuitive navigation.

Integrating with Enterprise Resource Planning (ERP), student information systems, and implementing Single Sign-On (SSO) with Microsoft domain passwords, can enhance operational efficiency and user experience. Here's how you can achieve these integrations:

1. Integration with ERP Systems:

SAP HCM is designed to seamlessly integrate with SAP ERP systems, allowing for unified data management across various business functions. This integration ensures that HR data is synchronized with other critical business processes, facilitating informed decision-making and streamlined operations.

SAP

2. Integration with Student Information Systems:

To connect SAP HCM with student information systems, you can utilize SAP's integration tools and middleware. These tools enable data exchange between systems, ensuring that student and HR information is consistent and up-to-date across platforms. Implementing such integrations may require custom development and thorough testing to meet specific institutional requirements.

3. Implementing Single Sign-On (SSO) with Microsoft Domain Passwords:

Enhancing user convenience and security can be achieved by integrating SAP HCM with Microsoft Entra ID (formerly Azure Active Directory) for SSO. This setup allows users to access SAP applications using their Microsoft domain credentials. The process involves:

Configuring SAP Systems for SSO: SAP systems can be configured to support SSO through SAML or OAuth protocols. This configuration enables SAP applications to trust Microsoft Entra ID as the identity provider.

MICROSOFT LEARN

Setting Up Trust Relationships: Establish a trust relationship between SAP Cloud Identity Services and Microsoft Entra ID. This involves exchanging metadata and configuring SAML settings to ensure secure authentication flows.

MICROSOFT LEARN

User Provisioning: Synchronize user accounts between Microsoft Entra ID and SAP HCM to ensure that user identities are consistent across systems. This synchronization can be achieved using provisioning tools that automate the creation and management of user accounts.

Part 2: Non-Functional Requirements

1. Security

* Use role-based access control (RBAC) to restrict access to sensitive data.
* Encrypt sensitive data in transit and at rest.
* Conduct regular vulnerability assessments and penetration testing.

2. Scalability

* Design the portal with modular architecture to support feature expansion.
* Use cloud-based infrastructure to handle growing user demands.

3. Usability

* Conduct user testing sessions to ensure intuitive design and functionality.
* Provide multilingual support (Arabic and English).

4. Performance

* Ensure response times of less than 2 seconds for 95% of user actions.
* Optimize mobile applications for low-bandwidth environments.

5. Maintenance

* Document all configurations, APIs, and workflows for ongoing maintenance.
* Schedule regular updates for feature enhancements and bug fixes.

Part 3: Implementation and Project Plan

Milestones

1. Phase 1: Requirements Gathering and Finalization
   * Deliverable: Approved requirements document.
2. Phase 2: Prototype Development
   * Deliverable: Interactive portal prototype and mobile app wireframes.
3. Phase 3: Core Feature Development
   * Deliverable: Fully functional portal with integrated core features.
4. Phase 4: Testing and Quality Assurance
   * Deliverable: Tested and debugged system ready for deployment.
5. Phase 5: Training and Go-Live
   * Deliverable: Trained staff and live intranet portal.

Part 2: Milestones:

Part 3: Responsibilities: