Digitalized Problem-Solving System

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Prepared by** | **Reviewed by** | **Approved by** |
| **Name** | Harshwardhan Patil | Rohit Chorghe | Mohammad Yusuf |
| Tanvi Bhosale | Tanvi Amlani |  |
| **Initials** | H.P. | R.C. | M.Y. |
|  | T.B. | T.A. |  |
| **Date** | 16/04/2024 | 18/04/2024 | 20/04/2024 |

**Submitted to**: Dr. Girish H. Subramanian.

**Team Members:**

Harshawardhan Patil

Rohit Chorghe

Tanvi Bhosale

Tanvi Amlani

Mohammad Yusuf

**TABLE OF CONTENT**

**INTRODUCTION**

PURPOSE OF PROJECT 4

      SCOPE OF THE PROJECT  4

      PROBLEM DEFINITION 6

      COMPETITIVE PRODUCTS/REFERENCES 6

FUNCTIONALITIES 7

**PROJECT MANAGEMENT**

      WORK AND TASK BREAKDOWN 9

      TIMELINE AND MILESTONES - GANTT CHART, PERT CHART 10

      ASSUMPTIONS, DEPENDENCIES, AND RISKS 11

**FUNCTIONAL SPECIFICATIONS**

      EXTERNAL INTERFACE SPECIFICATIONS 13

      FUNCTIONAL REQUIREMENTS 16

**HIGH LEVEL DESIGN**

      PURPOSE OF HIGH-LEVEL DESIGN 43

      SYSTEM ARCHITECTURE 43

      DESIGN CONSIDERATIONS AND CONSTRAINTS 46

**INTERFACE DESIGN**

      USER INTERFACE 52

**COMMON CONSIDERATIONS**

      SOFTWARE TEAM 58

      SOFTWARE PROCESS 59

      LIFECYCLE BREAKDOWN 59

      SOFTWARE PRODUCT DESCRIPTION 60

      SOFTWARE METRICS 61

ADDITIONAL REFERENCES 64

**INTRODUCTION**

**PURPOSE OF THE PROJECT**

In today's fast-paced and technologically advanced environment, efficient problem resolution is crucial for the seamless operation of businesses and organizations. In response to the imperative need for swift and error-free problem resolution in today's dynamic business environment, this project endeavors to design, develop, and implement a digitalized ticket system, aiming to revolutionize and streamline the process of addressing issues. Key objectives include enhancing efficiency through the automation of problem reporting, analysis, and resolution, minimizing human errors in ticket creation through validation checks, providing stakeholders with real-time visibility into issue statuses through automated updates and notifications, establishing a centralized data management system for easy retrieval and analysis of historical ticketing data, and designing a user-friendly interface to cater to users with varying technical expertise. The successful execution of this digitalized ticket system is anticipated to not only overcome current challenges linked with manual ticketing but also significantly contribute to heightened overall efficiency, increased customer satisfaction, and enhanced organizational productivity.

**SCOPE OF THE PROJECT**

This software will have two major components: users and site administrators.

**Users:**

The HR portal, integrated with the company's database, automatically enrolls employees from specific departments, ensuring streamlined access to essential support services. Upon successful registration, employees are required to log in to the portal using their unique employee ID or username, coupled with a password. To fortify security, a two-step authentication process is mandatory for every login attempt.

Once logged in, the portal offers a user-friendly interface allowing employees to efficiently manage and track their support requests such as:

1. Dashboard Overview: Quick access to current activities, including open and historical tickets.

2. Advanced Search and Filter Options: Easy-to-use tools to locate specific tickets based on various criteria.

3. Ticket Progress Tracking: Real-time updates with a dynamic progress bar for tracking ticket resolution status.

4. New Ticket Creation: Simple process for initiating new support requests, with options for attachments and categorization.

5. Active Ticket Editing: Capability to update or amend details on active tickets as situations evolve.

6. Reactivation of Completed Tickets: Allows users to reopen resolved tickets if issues persist, with automatic information carryover for continuity.

**Administrators**:

Site administrators are pivotal to the seamless operation and oversight of the entire ticket tracking system. They wield comprehensive authority, enabling them to not only view but also edit any ticket within the system, ensuring that issues are addressed promptly and efficiently. Their role extends to controlling access to the system; no new employee can be added without their explicit authorization, which is granted following a formal request. This gatekeeping ensures that only authorized personnel have access to the system, maintaining its integrity and security.

In addition to these responsibilities, site administrators have exclusive access to sensitive and confidential data, including the integrated platform database. This access is critical for safeguarding user information and maintaining the system's confidentiality. The administrators' duties also encompass the technical integration of the site with other websites and APIs, a task that involves negotiating and setting up connections with external platforms to enhance the system's capabilities and ensure smooth interoperability.

Beyond these core responsibilities, site administrators are also tasked with maintaining the system's infrastructure. This includes monitoring system performance to preemptively identify and address potential issues, ensuring the system's stability and reliability. They are responsible for the deployment of new features and updates, carefully managing the rollout of enhancements to improve functionality without disrupting the user experience. Furthermore, they play a crucial role in strategizing for disaster recovery, developing robust plans to mitigate data loss and ensure rapid system restoration in the event of an outage.

Moreover, administrators are instrumental in fostering a secure environment, implementing and updating security protocols to protect against cyber threats and unauthorized access. Their comprehensive oversight ensures that the ticket tracking system not only operates flawlessly but also remains secure, reliable, and aligned with the evolving needs of its users. Through their expertise and vigilance, site administrators are the linchpins of the system, guaranteeing that it continues to serve as an indispensable tool for efficient issue tracking and resolution.

**PROBLEM DEFINITION**

The current manual ticketing system suffers from several critical inefficiencies, resulting in delays and a notable lack of transparency. These issues manifest in frequent miscommunication, delayed responses to problem reports, and inaccuracies in data processing, collectively impeding the overall effectiveness of problem resolution within the system. The challenges associated with the existing system not only led to frustrated end-users but also pose significant obstacles to the timely and accurate resolution of reported issues. These inefficiencies highlight a pressing need for a comprehensive overhaul of the ticketing process. The purpose of this project is to confront and rectify these challenges by introducing a modernized, digitalized ticket system that adheres to contemporary technological standards. By doing so, the project aims to streamline and revolutionize the problem resolution process, addressing the root causes of inefficiency, delays, and a lack of transparency inherent in the current manual ticketing system.

**COMPETITIVE PRODUCTS/REFERENCES:**

ServiceNow is a well-known rival in the market for IT firms that provide ticketing systems. A variety of IT service management (ITSM) solutions are offered by ServiceNow, such as ticketing systems for managing requests, incidents, modifications, and other IT-related duties. Jira Service Management (previously known as Jira Service Desk) from Atlassian, Zendesk, Freshdesk, BMC Helix ITSM, and Cherwell Service Management are some of the other competitors in this market. To help IT departments and organizations manage and handle issues and requests more effectively, these companies provide a range of tools and functionalities.

**FUNCTIONALITIES**:

The ticket tracking system is designed with a comprehensive set of features to streamline the ticket management process, enhance collaboration, and improve issue resolution efficiency. Here’s a detailed overview of its functionalities:

1. Ticket Submission: Users can initiate support requests by completing all necessary fields across various sections, ensuring detailed documentation of the issue.

2. Ticket Access Control: The system allows users to set preferences for who can view or edit tickets, enhancing privacy and collaboration as needed.

3. Priority Assignment and Delegation: Within the ticket submission form, users specify the issue's priority level (1 to 5) and can assign it either to a random developer within the department or to a specific individual, optimizing workflow and resource allocation.

4. Progress Saving: As tickets are created, users have the option to manually save their progress. Additionally, the system automatically saves progress at the end of each section, reducing data loss risk.

5. Concurrency Management: The system detects when a ticket is being edited from multiple devices simultaneously, alerting subsequent users with a pop-up message that editing is already in progress elsewhere, thus preventing data inconsistency.

6. Evidence Upload: Users can attach up to five documents or photos as evidence to support their ticket, with the system automatically selecting the two most pertinent ones for inclusion in the final report based on relevance.

7. Digital Report Generation: Upon completion, a digital report is automatically generated, summarizing the ticket's details, including selected evidence and relevant trend charts, presenting a concise overview of the issue for efficient resolution.

8. Trend Analysis: The system analyzes data from previously resolved tickets within the department to generate trend charts, highlighting key impact factors and aiding in the identification of recurring issues or patterns.

9. Resolution Tracking: A visual indicator displays the current resolution stage of each ticket, offering users real-time updates on progress.

10. Email Notifications: The system facilitates communication by automatically generating email notifications for significant events, such as ticket assignments or updates on progress, keeping all relevant parties informed.

11. Ticket Modification: Active tickets can be edited, albeit with limitations on certain fields to preserve the integrity of the original issue report.

12. Ticket Reactivation: Users have the ability to view and, if necessary, reactivate previously resolved or inactive tickets as new issues, ensuring that recurring problems are promptly addressed.

13. Advanced Search Functionality: The platform includes a powerful search feature with multiple filters, allowing users to find tickets based on a variety of criteria, such as department, category, or region.

14. Automatic User Onboarding: New employees are automatically granted system access through integration with the department's database, such as HRPortal, with updates occurring on a weekly basis, ensuring that all team members have timely access to the system.

These features collectively enhance the efficiency, transparency, and responsiveness of the ticket tracking system, making it an invaluable tool for managing support requests and fostering a collaborative work environment.

**PROJECT MANAGEMENT**

**WORK AND TASK BREAKDOWN**

**1)** **Project Definition Phase:**

* Defining purpose of the project
* Establishing the project scope
* Identifying potential project challenges
* Establishing basic and functional requirements

**2)** **Project Planning Phase:**

* Breakdown of all essential tasks
* Creating a tentative timeline of the activities
* Identifying any assumptions, dependencies, and potential risks
* Estimate the cost required and the time for project completion

**3)** **Project Design Phase:**

* Define essential use cases
* Draw flowcharts to demonstrate their utilization
* Define adequate functional specifications
* Design System Architecture

**4)** **Project Development Phase:**

* Develop Web Server
* Develop Client UI

**5)** **Project Testing Phase:**

* Unit testing
* Integration testing
* Functional testing
* Security testing
* Stress testing

**TIMELINE AND MILESTONES:**

**GANTT CHART**

**ASSUMPTIONS, DEPENDENCIES AND RISKS :**

**Assumptions:**

* Technical Infrastructure: Assumes the organization possesses the necessary technical infrastructure for deployment, including hardware, network capabilities, and compatibility with existing systems.
* Digital Literacy: Assumes users have sufficient digital literacy to interact effectively with the new system, including troubleshooting minor issues.
* Reliable Internet Connectivity: Presumes constant, reliable internet access for cloud-based functionalities, data access, and system updates.
* Adaptability and Compatibility: Assumes the current organizational processes and systems are adaptable to integrate smoothly with the digitalized system without significant overhauls.

**Dependencies:**

* Skilled Developers: Relies on the availability of skilled software developers and engineers proficient in the technologies used for building and maintaining the system.
* Existing Systems Access: Depends on access to the organization’s existing databases and IT systems for integration, data migration, and synchronization.
* Third-party Services: May depend on third-party services or APIs for specific functionalities, such as payment processing, data analytics, or external communications.
* Stakeholder Support: Requires ongoing support and buy-in from key stakeholders within the organization, including management, IT staff, and end-users.

**Risks:**

* Technical Challenges: The risk of encountering unforeseen technical challenges that could delay development, increase costs, or necessitate compromises in functionality.
* User Resistance: Potential resistance from users accustomed to existing processes, which could hinder adoption and require additional training and support efforts.
* Security Vulnerabilities: Introducing a new system could expose the organization to new security vulnerabilities, requiring rigorous testing and ongoing vigilance.
* Project Overruns: The risk of the project exceeding its timeline and budget, which could strain resources and impact the organization’s operations.
* Integration Complexities: Challenges in integrating the new system with existing systems could lead to operational disruptions, data inconsistencies, or additional costs.

**FUNCTIONAL SPECIFICATIONS**

This section provides a detailed description of the common considerations and expectations across all functional requirements for the digitalized problem-solving report/system.

**EXTERNAL INTERFACE SPECIFICATIONS**

**User Interface**

The system offers a highly intuitive and user-friendly web interface that is accessible through all major web browsers, including Chrome, Firefox, Safari, and Edge, on both desktop and mobile devices. The interface is designed to be responsive, ensuring optimal viewing and interaction across various screen sizes and resolutions. The layout and navigation are carefully crafted to provide a seamless and efficient user experience, with clear and concise labeling, logical grouping of functions, and consistent visual elements. The interface incorporates accessibility best practices, such as keyboard navigation and proper color contrast, to accommodate users with disabilities.

**Hardware Interface**

The system is hosted on a cluster of high-performance servers with redundant components to ensure maximum uptime and reliability. The servers are equipped with state-of-the-art processors, ample memory, and fast solid-state drives to handle a large volume of concurrent users and ticket data processing. The server infrastructure is scalable, allowing for dynamic resource allocation based on system load and performance requirements. Regular data backups are performed to multiple secure locations, ensuring data integrity and quick recovery in case of any hardware failures. The system also utilizes content delivery networks (CDNs) to distribute static assets and improve loading times for users across different geographical locations.

**Software Interface**

The system is developed using a modern technology stack, following industry best practices and standards for web application development. The frontend is built using HTML5, CSS3, and JavaScript, with the help of responsive web design frameworks like Bootstrap or Material Design to ensure cross-browser compatibility and consistent visual appearance. The backend is powered by a robust and scalable server-side language, such as Java SpringBoot, coupled with a high-performance database system like SQL for efficient data storage and retrieval. RESTful APIs are used for seamless communication between the frontend and backend components. The system architecture follows a modular and loosely coupled design, promoting code reusability, maintainability, and extensibility. Proper security measures, such as input validation, authentication, and authorization mechanisms, are implemented to protect against common web vulnerabilities. The codebase is version-controlled using Git, and automated testing and deployment processes are set up to ensure code quality and streamline the software development lifecycle.

**Performance Requirements**

The system is optimized for high performance and can handle a large number of concurrent users and ticket submissions without compromising response times or user experience. The server infrastructure is designed to scale horizontally, allowing for the addition of more servers to handle increased traffic loads. Caching mechanisms are implemented at various layers to reduce database queries and improve overall system performance. The system is capable of processing at least 1,000 ticket submissions per minute, with an average response time of under 200 milliseconds for most user actions. The system is designed to handle a minimum of 10,000 concurrent users without significant performance degradation. Load balancing techniques are employed to distribute the workload evenly across multiple servers, ensuring optimal resource utilization and minimizing the risk of system bottlenecks. Regular performance monitoring and profiling are conducted to identify and optimize any performance bottlenecks, ensuring a consistently fast and responsive user experience.

**Attributes**

* **Availability**: The system is designed to provide 99.99% uptime, ensuring that users can access and interact with the system at any time, from anywhere. Multiple redundant servers are deployed across different geographic regions to minimize the impact of any single server failure. Automatic failover mechanisms are in place to seamlessly redirect traffic to healthy servers in case of any issues. Regular server maintenance and updates are scheduled during off-peak hours to minimize disruption to users. Comprehensive monitoring and alerting systems are implemented to proactively detect and resolve any potential availability issues before they impact users.
* **Security**: The system employs a multi-layered security approach to protect sensitive ticket data and user information from unauthorized access and breaches. All data transmitted between the user's browser and the server is encrypted using industry-standard HTTPS protocol with strong SSL/TLS certificates. User passwords are hashed and salted using secure algorithms like bcrypt or PBKDF2 before storing them in the database. Role-based access control (RBAC) is implemented to ensure that users can only access and perform actions on tickets and data relevant to their assigned roles and permissions. Regular security audits and penetration testing are conducted by independent third-party experts to identify and address any potential vulnerabilities. Strict data retention and deletion policies are enforced to comply with relevant data protection regulations, such as GDPR or HIPAA.
* **Reliability**: The system undergoes rigorous testing and quality assurance processes to ensure reliable performance and data integrity. Comprehensive unit, integration, and system testing are performed to verify the functionality and reliability of individual components and the system as a whole. Automated error handling and logging mechanisms are implemented to capture and report any errors or exceptions in real-time, enabling quick identification and resolution of issues. Data backups are regularly performed and securely stored in multiple locations to protect against data loss due to hardware failures or disasters. Fault tolerance and disaster recovery mechanisms are in place to ensure that the system can recover from failures and resume normal operation with minimal downtime.

**FUNCTIONAL REQUIREMENTS**

**Requirement ID: DPSR-RN-001-01-01**

**Process ID: 3.2, 3.9, 3.10**

**Ticket Submission**

**Operating Specification:**

Users can initiate support requests by filling out a comprehensive ticket submission form (Process 3.2). The form includes the following fields:

* Title (mandatory): A concise and descriptive title for the issue or request.
* Description (mandatory): A detailed explanation of the problem, including steps to reproduce, expected behavior, and actual behavior.
* Category (mandatory): A dropdown list of predefined categories to classify the ticket, such as "Hardware," "Software," "Network," "Security," "Department," "Sub-department," etc.
* Priority (mandatory): A dropdown list to indicate the urgency of the ticket, with options like "Low," "Medium," "High," and "Critical."
* Attachments (optional): Users can attach up to five files, such as screenshots, log files, or documents, to provide additional context or evidence for the ticket. Supported file formats include JPG, PNG, PDF, and TXT, with a maximum file size of 10MB per attachment.
* Assigned To (optional): Users can select an individual or team to assign the ticket to from a list of available options based on their department and expertise.
* Due Date (optional): Users can specify a desired resolution date for the ticket.

The ticket submission form includes client-side validation to ensure that all mandatory fields are filled out and that the entered data meets the specified format and constraints.

Upon successful submission of the form, the system removes any draft tickets associated with the user (Process 3.9) and updates the database with the newly created ticket (Process 3.10). The system generates a unique ticket ID and sends a confirmation email to the user with the ticket details and ID for future reference. The ticket is then stored in the system's database, along with its associated metadata, such as the submission timestamp, submitter's user ID, and initial status (e.g., "New" or "Open").

**Requirement ID: DPSR-RN-002-01-02**

**Process ID: 2.1, 3.3**

**Ticket Access Control**

**Operating Specification:**

The system provides granular access control mechanisms to ensure that tickets are only accessible to authorized users based on their roles and permissions.

Upon successful login (Process 2.1), the system applies access control checks (Process 3.3) to determine the user's permissions and access rights within the ticketing system. Each ticket has the following access control options:

* View Permissions: Determines who can view the ticket and its associated details. Options include.
  + Private: Only the ticket submitter and assigned individual/team can view the ticket.
  + Department: All members of the assigned department can view the ticket.
  + Organization: All users within the organization can view the ticket.
  + Public: The ticket is publicly accessible to anyone with the ticket URL (used for external communication).
* Edit Permissions: Determines who can modify the ticket details, add comments, or change the ticket status. Options include:
  + Submitter: Only the ticket submitter can edit the ticket.
  + Assigned: Only the assigned individual/team can edit the ticket.
  + Department: All members of the assigned department can edit the ticket.

The default access control settings are configurable by system administrators and can be set based on the organization's security policies and collaboration requirements.

Access control checks are performed server-side to ensure that users can only view or modify tickets they are authorized to access. If a user attempts to access a ticket they are not permitted to view or edit, the system displays an "Access Denied" message and logs the attempt for auditing purposes.

**Requirement ID: DPSR-RN-003-01-03**

**Process ID: 3.4**

**Priority Assignment and Delegation**

**Operating Specification:**

During the ticket submission process (Process 3.4), users can assign a priority level to indicate the urgency and importance of the issue. The available priority levels are:

* Low: The issue is minor and does not significantly impact productivity or functionality.
* Medium: The issue affects productivity or functionality but has reasonable workarounds available.
* High: The issue significantly impacts productivity or functionality and requires prompt attention.
* Critical: The issue is severely impacting business operations or causing system downtime and requires immediate resolution.

Users can also choose to assign the ticket to a specific individual or team within the relevant department. The system provides a searchable dropdown list of available assignees, populated based on the user's department and the ticket category. If no assignee is selected, the system automatically assigns the ticket to the next available developer in the department based on their current workload and expertise.

**Requirement ID: DPSR-RN-004-01-04**

**Process ID: 3.5**

**Progress Saving**

**Operating Specification:**

The system provides a seamless and user-friendly experience for ticket creation by allowing users to save their progress at any point during the ticket submission process (Process 3.5). The progress saving feature includes:

* Automatic Saving: The system automatically saves the user's progress as they fill out the ticket submission form. The saving occurs in the background at regular intervals (e.g., every 30 seconds) without disrupting the user's workflow. This ensures that users don't lose their entered data in case of unexpected browser crashes or network interruptions.
* Manual Saving: In addition to automatic saving, users have the option to manually save their progress by clicking a "Save Draft" button provided on the ticket submission form. This allows users to proactively save their work before navigating away from the page or taking a break.
* Draft Persistence: The system stores the saved draft tickets in the user's browser local storage or server-side database, depending on the implementation. The saved drafts are associated with the user's account and can be accessed and resumed from any device or browser session.
* Draft Retrieval: When a user starts a new ticket submission, the system checks for any existing draft tickets associated with their account. If a draft is found, the system prompts the user to either resume the draft or start a new ticket. If the user chooses to resume the draft, the ticket submission form is pre-populated with the previously saved data, allowing the user to continue from where they left off.
* Draft Management: Users can view and manage their draft tickets from a dedicated "Drafts" section within the system. The drafts are listed in chronological order, with the most recent draft at the top. Users can edit, delete, or submit the draft tickets as needed.
* Auto-saving Indicators: To provide visual feedback to users, the system displays auto-saving indicators or status messages near the ticket submission form. These indicators inform users when their progress is being saved automatically and when the saving process is complete.
* Conflict Resolution: In the rare case where multiple draft versions of the same ticket exist (e.g., due to concurrent editing from different devices), the system applies a conflict resolution mechanism. It prompts the user to choose which version to keep and merges the changes if possible, ensuring data consistency and preventing accidental data loss.

The progress saving feature enhances the user experience by preventing data loss, allowing users to resume their work seamlessly, and providing peace of mind during the ticket submission process.

**Requirement ID: DPSR-RN-005-01-05**

**Process ID: 3.6**

**Concurrency Management**

**Operating Specification:**

The system implements robust concurrency management mechanisms (Process 3.6) to handle scenarios where multiple users attempt to edit the same ticket simultaneously from different devices or browser sessions. The concurrency management feature ensures data integrity, prevents conflicting updates, and provides a smooth collaboration experience for users.

* Locking Mechanism: When a user starts editing a ticket, the system applies a locking mechanism to prevent other users from making concurrent modifications. The lock is acquired when the user clicks the "Edit" button or starts typing in any of the ticket fields.
* Lock Visibility: While a ticket is being edited, the system displays a prominent visual indicator (e.g., a lock icon or a "Currently being edited" message) to inform other users that the ticket is locked for editing. This helps avoid confusion and prevents users from attempting to edit the ticket simultaneously.
* Lock Timeout: To prevent indefinite locking of tickets, the system implements a lock timeout mechanism. If the user who acquired the lock remains idle or inactive for a specified period (e.g., 10 minutes), the lock is automatically released, allowing other users to edit the ticket. The lock timeout duration is configurable by system administrators.
* Notification and Refresh: If a user attempts to edit a ticket that is already being edited by another user, the system displays a notification message informing them that the ticket is currently locked. The notification includes the name of the user who is currently editing the ticket and the time remaining until the lock expires. The system also provides an option to refresh the ticket view to check if the lock has been released.
* Collaborative Editing: In cases where real-time collaboration on tickets is required, the system can implement collaborative editing features. This allows multiple users to edit different parts of the ticket simultaneously, with changes being synced in real-time across all collaborators. Collaborative editing can be enabled or disabled based on the ticket's access control settings and the organization's collaboration needs.
* Audit Trail: The system maintains a detailed audit trail of all ticket modifications, including the user who made the change, the timestamp, and the specific changes made. This audit trail helps in tracking the history of the ticket and provides accountability for the modifications.

The concurrency management feature ensures that tickets are edited in a controlled and collaborative manner, preventing data inconsistencies and conflicts arising from simultaneous modifications by multiple users.

**Requirement ID: DPSR-RN-006-01-06**

**Process ID: 3.7**

**Evidence Upload**

**Operating Specification:**

The system allows users to attach relevant evidence files, such as documents, images, or videos, to support their ticket submissions (Process 3.7). The evidence upload feature is designed to be user-friendly, efficient, and secure.

* File Upload Interface: The ticket submission form includes a dedicated section for evidence upload. Users can click on an "Attach Evidence" button or drag and drop files directly onto the designated area to initiate the upload process.
* File Format and Size Limitations: The system supports a wide range of common file formats for evidence upload, including PDF, DOCX, XLSX, JPG, PNG, and MP4. The maximum file size allowed per evidence file is configurable by system administrators, typically set to a reasonable limit (e.g., 20MB) to ensure efficient upload and storage.
* Multiple File Upload: Users can attach up to five evidence files per root cause. The system provides a clear indication of the number of files selected and the total size of the uploaded files. If the user attempts to upload more than the allowed number of files or exceeds the size limit, the system displays an appropriate error message and guides the user to make the necessary adjustments.
* File Validation and Security: The system performs server-side validation and security checks on the uploaded evidence files to ensure they are of the allowed file types and within the size limits. The uploaded files are scanned for viruses and malware using industry-standard antivirus software to prevent any potential security threats.
* Automatic File Renaming: To maintain uniqueness and avoid file naming conflicts, the system automatically renames the uploaded evidence files using a predefined naming convention. The naming convention typically includes the ticket ID, a timestamp, and a sequential number (e.g., TICKET-1234\_EVIDENCE-20230524-1.pdf).
* Evidence Storage and Retrieval: The uploaded evidence files are securely stored on the server, typically in a dedicated storage location or database. The system maintains the association between the evidence files and the corresponding ticket. Users can view and download the attached evidence files from the ticket details page.
* Evidence Confidentiality: The system ensures the confidentiality of the uploaded evidence files by implementing access control mechanisms. Only users with the appropriate permissions, such as the ticket submitter, assigned personnel, and authorized administrators, can view and download the evidence files associated with a ticket.
* Audit and Retention: The system maintains an audit log of all evidence file uploads, including details such as the user who uploaded the file, the timestamp, and the file metadata. The evidence files are retained in accordance with the organization's data retention policies and can be archived or deleted as per the defined schedule.

The evidence upload feature enhances the ticket submission process by allowing users to provide supporting materials that help in understanding and resolving the reported issues effectively.

**Requirement ID: DPSR-RN-007-01-07**

**Process ID: 3.8**

**Digital Report Generation**

**Operating Specification:**

Upon successful submission of a ticket, the system automatically generates a comprehensive digital report (Process 3.8) summarizing the ticket details, selected evidence files, and relevant trend charts. The digital report serves as a concise and informative overview of the issue, facilitating efficient communication and resolution.

* Report Template: The system utilizes a predefined report template that ensures consistency and professionalism across all generated reports. The template includes sections for ticket metadata, description, selected evidence files, and trend charts.
* Ticket Metadata: The report includes key ticket metadata such as the ticket ID, submission date and time, submitter's name and contact information, assigned personnel or team, priority level, and current status.
* Ticket Description: The report provides a clear and detailed description of the reported issue, including the steps to reproduce, expected behavior, and actual behavior. The description is formatted for readability, with proper headings, bullet points, and styling as needed.
* Selected Evidence Files: The two most relevant evidence files, as determined by the system's relevance analysis or manually selected by the user, are included in the report. The evidence files are embedded or attached to the report in a suitable format (e.g., PDF, image) for easy access and review.
* Trend Charts: The system generates visual trend charts based on historical ticket data to provide insights into recurring issues, common impact factors, and resolution patterns. The trend charts may include:
  + Issue Frequency: A bar chart showing the number of tickets reported for similar issues over a specified time period.
  + Impact Analysis: A pie chart illustrating the distribution of impact levels (e.g., low, medium, high) for tickets related to the reported issue.
  + Resolution Time: A line graph depicting the average resolution time for tickets of similar nature, helping to set expectations for resolution timeframes.
* Branding and Customization: The digital report incorporates the organization's branding elements, such as logo, color scheme, and typography, to maintain a professional and consistent look and feel. The report template can be customized by system administrators to align with the organization's specific reporting requirements.
* Report Generation and Delivery: The digital report is generated in a widely compatible format, such as PDF, to ensure easy distribution and accessibility across different devices and platforms. The report is automatically attached to the ticket and sent to the submitter and assigned personnel via email. A copy of the report is also stored in the system's database for future reference and auditing purposes.
* Report Versioning: If updates or modifications are made to the ticket after the initial report generation, the system maintains version control and generates updated reports as needed. The updated reports include a clear indication of the changes made and the date of the update.
* Report Analytics: The system tracks and analyzes the generated reports to identify trends, common issues, and areas for improvement. The analytics data can be used to optimize the ticket resolution process, allocate resources effectively, and enhance the overall performance of the problem-solving system.

The digital report generation feature streamlines the ticket submission process by automatically creating a comprehensive and informative summary of the reported issue. The report serves as a valuable communication tool, enabling all stakeholders to have a clear understanding of the problem and facilitating efficient resolution.

**Requirement ID: DPSR-RN-008-01-08**

**Process ID: 4.1, 4.2, 4.3**

**Trend Analysis**

**Operating Specification:**

The system incorporates advanced trend analysis capabilities to identify patterns, recurring issues, and key impact factors from historical ticket data. The trend analysis feature helps organizations gain valuable insights, optimize their problem-solving processes, and proactively address potential issues.

* Data Collection (Process 4.1): The system collects and stores relevant ticket data, including ticket metadata, description, category, priority, resolution time, and associated evidence files. The data is securely stored in a structured format, such as a relational database or data warehouse, to facilitate efficient analysis.
* Data Preprocessing (Process 4.2): Before performing trend analysis, the system preprocesses the collected ticket data to ensure data quality and consistency. This involves tasks such as data cleansing (e.g., removing duplicates, handling missing values), data normalization (e.g., standardizing date formats, converting units), and data transformation (e.g., extracting relevant features, categorizing text data).
* Trend Identification (Process 4.3): The system employs various statistical and machine learning techniques to identify trends and patterns in the ticket data. Some of the techniques used include:
  + Time Series Analysis: The system analyzes ticket data over time to identify seasonal patterns, trends, and anomalies. It considers factors such as ticket volume, resolution time, and impact levels to detect any significant changes or shifts in the data.
  + Clustering: The system applies clustering algorithms (e.g., k-means, hierarchical clustering) to group similar tickets based on their characteristics, such as category, description, or impact level. This helps identify common issues and their associated attributes.
  + Association Rule Mining: The system uses association rule mining techniques to discover frequent co-occurrences and correlations between different ticket attributes. For example, it can identify if certain categories of tickets are more likely to have a high priority or longer resolution times.
  + Text Mining: The system applies natural language processing (NLP) techniques to analyze the textual data in ticket descriptions and comments. It extracts key phrases, sentiment, and topics to identify common themes and recurring issues.
* Visualization: The system generates intuitive and visually appealing trend charts, graphs, and dashboards to present the identified trends and patterns. The visualizations may include:
  + Line Charts: Showing the trend of ticket volume, resolution time, or impact levels over time.
  + Bar Charts: Comparing the frequency or counts of different categories, priority levels, or impact factors.
* Pie Charts: Illustrating the distribution of ticket attributes, such as category or status.
  + Heatmaps: Highlighting the correlation or co-occurrence of different ticket attributes.
* Interactive Exploration: The system provides an interactive interface for users to explore and drill down into the trend data. Users can filter the data based on various criteria (e.g., date range, category, priority), zoom in/out of specific time periods, and hover over data points to view detailed information.

The trend analysis feature empowers organizations to leverage their historical ticket data to gain valuable insights, identify recurring issues, and optimize their problem-solving processes. By proactively addressing emerging trends and patterns, organizations can improve efficiency, reduce resolution times, and enhance overall customer satisfaction.

**Requirement ID: DPSR-RN-009-01-09**

**Process ID: 6.1, 6.2, 6.3**

**Resolution Tracking**

**Operating Specification:**

The system provides a robust resolution tracking feature that allows users to easily monitor the progress and status of their submitted tickets. The feature includes visual indicators, real-time updates, and detailed information to keep users informed throughout the ticket lifecycle.

* Ticket Status (Process 6.2): Each ticket is assigned a status that reflects its current state in the resolution process. The common ticket statuses include:
  + Open: The ticket has been submitted and is awaiting initial review or assignment.
  + In Progress: The ticket has been assigned and is actively being worked on by the responsible team or individual.
  + On Hold: The ticket is temporarily paused due to pending information, external dependencies, or resource constraints.
  + Resolved: The ticket has been successfully resolved, and the reported issue has been addressed.
  + Closed: The ticket has been resolved, and no further action is required. It has been formally closed by the assigned team or individual.
* Visual Indicators: The system uses intuitive visual indicators to represent the current status of each ticket. These indicators may include color-coded labels, icons, or progress bars that clearly communicate the ticket's state at a glance. For example:
  + Open tickets may have a red label or icon.
  + In Progress tickets may have a yellow label or a partially filled progress bar.
  + Resolved tickets may have a green label or a checkmark icon.
* Ticket Details Page (Process 6.1): When a user clicks on a ticket from their ticket list or dashboard, they are directed to a detailed ticket information page. This page provides comprehensive information about the ticket, including:
  + Ticket metadata: Ticket ID, submission date and time, category, priority, and assigned team or individual.
  + Ticket description: The original description provided by the user during ticket submission.
  + Comments and Updates: A chronological list of all comments, updates, and actions taken on the ticket by the assigned team or individual.
  + Attachments: Any evidence files or supporting documents attached to the ticket.
  + Resolution Details (Process 6.3): Once the ticket is resolved, the resolution details, including the resolution date, the team or individual who resolved it, and any relevant notes or explanations, are displayed.
* Real-time Updates: The system provides real-time updates on the ticket status and progress. Whenever there is a change in the ticket status or a new comment is added, the user is notified through various channels, such as:
  + In-app Notifications: The user receives a notification within the problem-solving system interface, typically in the form of a pop-up message or a badge on the ticket icon.
  + Email Notifications: The user receives an email notification with the updated ticket information and a link to view the ticket details in the system.
* Ticket History: The system maintains a complete history of all actions, updates, and changes made to a ticket. Users can access the ticket history to view the chronological sequence of events, including status changes, comments, assignments, and resolution details. The ticket history provides transparency and accountability, allowing users to track the progress and understand the steps taken to resolve their issues.
* Ticket Search and Filtering: The system offers powerful search and filtering capabilities to help users quickly find and track specific tickets. Users can search for tickets based on various criteria, such as ticket ID, title, description, category, priority, status, or date range. They can also apply filters to narrow down the ticket list based on their preferences.
* Resolution Metrics: The system calculates and displays relevant resolution metrics for each ticket, such as:
  + Time to First Response: The time elapsed between the ticket submission and the first response from the assigned team or individual.
  + Time to Resolution: The total time taken from the ticket submission to its resolution.
  + Resolution SLA Compliance: An indicator showing whether the ticket was resolved within the predefined Service Level Agreement (SLA) timeframe.
* Feedback and Satisfaction: After a ticket is resolved, the system may prompt the user to provide feedback and rate their satisfaction with the resolution process. This feedback helps the organization assess the effectiveness of their problem-solving efforts and identify areas for improvement.
* Escalation and Notification: If a ticket remains unresolved or unattended for a specified period, the system automatically escalates it to higher levels of management or supervisors. Escalation rules and thresholds can be configured based on the ticket priority, category, or SLA compliance. The system sends notifications to the appropriate stakeholders, ensuring that critical issues are promptly addressed.

The resolution tracking feature enhances user satisfaction and confidence in the problem-solving system by providing transparent, real-time updates and detailed information throughout the ticket lifecycle. It enables users to stay informed, track the progress of their issues, and hold the responsible teams accountable for timely and effective resolutions.

**Requirement ID: DPSR-RN-010-01-10**

**Process ID: 5.1, 5.2, 5.3**

**Email Notifications**

**Operating Specification:**

The system includes a comprehensive email notification feature that keeps users and relevant stakeholders informed about important ticket events and updates. The email notifications ensure that all parties involved in the problem-solving process are kept in the loop and can take appropriate actions in a timely manner.

* Notification Triggers (Process 5.1): The system sends email notifications based on predefined triggers or events related to the ticket lifecycle. Some common triggers include:
  + New Ticket Submission: When a user submits a new ticket, an email notification is sent to the user confirming the receipt of their ticket and providing the ticket ID for future reference.
  + Ticket Assignment: When a ticket is assigned to a specific team or individual, an email notification is sent to the assignee, informing them about the new ticket and providing a link to view the ticket details.
  + Ticket Status Change: Whenever the status of a ticket changes (e.g., from Open to In Progress, from In Progress to Resolved), an email notification is sent to the user and the assigned team or individual, updating them about the new status.
  + New Comment or Update: When a new comment or update is added to a ticket by either the user or the assigned team, an email notification is sent to all relevant parties, ensuring they are aware of the latest developments.
  + Ticket Resolution: When a ticket is marked as resolved, an email notification is sent to the user, informing them that their issue has been addressed and providing any necessary resolution details.
* Notification Templates (Process 5.2): The system uses predefined email templates for different notification triggers to ensure consistency and clarity in communication. The templates include placeholders for dynamic information, such as the ticket ID, subject, description, status, and assignee. The templates are customizable by system administrators to align with the organization's branding and communication guidelines.
* Recipient Management: The system maintains a list of relevant recipients for each ticket based on their roles and involvement. The recipients may include:
  + Ticket Submitter: The user who submitted the ticket.
  + Assigned Team or Individual: The team or individual responsible for resolving the ticket.
  + Escalation Contacts: Higher-level managers or supervisors who need to be informed about critical or overdue tickets.
* Notification Content (Process 5.3): The email notifications include essential information about the ticket, such as:
  + Ticket ID and Subject: The unique identifier and subject line of the ticket.
  + Ticket Description: A summary or excerpt of the ticket description.
  + Ticket Status: The current status of the ticket (e.g., Open, In Progress, Resolved).
  + Assignee: The team or individual assigned to resolve the ticket.
  + Comments or Updates: Any new comments or updates added to the ticket.
  + Resolution Details: In case of a resolution notification, the details of the resolution, including any steps taken or recommendations provided.
* Notification Links: The email notifications include direct links to the corresponding ticket in the problem-solving system. Users can click on these links to quickly access the ticket details, view the full history, and take any necessary actions.
* Notification Security: The system ensures the security of the email notifications by:
  + Encrypting the email content and any sensitive information.
  + Verifying the recipient's email address to prevent unauthorized access.
  + Including a disclaimer or footer in the email, reminding users to keep the information confidential and not share it with unauthorized parties.
* Notification Opt-Out: The system provides an option for users to opt-out of specific types of notifications or unsubscribe from all notifications if they prefer. The opt-out process is clearly communicated in the email notifications and can be easily managed by users through their notification preferences.

The email notification feature enhances communication and collaboration throughout the problem-solving process. It ensures that all relevant parties are promptly informed about ticket updates, assignments, and resolutions, reducing delays and improving overall efficiency. The notifications also serve as a record of important ticket events and help maintain accountability and transparency within the system.

**Requirement ID: DPSR-RN-011-01-11**

**Process ID: 7.1, 7.2, 7.3, 7.4**

**Ticket Modification**

**Operating Specification:**

The system allows users to modify and update their submitted tickets to provide additional information, clarify details, or request changes. The ticket modification feature is designed to facilitate effective communication and collaboration between users and the assigned teams or individuals.

* Modification Permissions (Process 7.1): The system enforces strict access controls and permissions for ticket modification. Only authorized users, such as the ticket submitter and the assigned team or individual, are allowed to modify the ticket details. Other users with view-only access to the ticket cannot make any changes.
* Editable Fields (Process 7.2): The system defines specific fields within the ticket that are editable by users. These fields may include:
  + Ticket Subject: Users can modify the subject line of the ticket to better reflect the nature of the issue or request.
  + Ticket Description: Users can update or append additional information to the ticket description to provide more context, clarify requirements, or report any changes in the situation.
  + Ticket Category: Users may be allowed to change the category of the ticket if they realize it was initially misclassified or if the nature of the issue has evolved.
  + Ticket Priority: In some cases, users may be permitted to adjust the priority level of the ticket based on the urgency or impact of the issue.
* Revision History (Process 7.3): The system keeps track of all modifications made to the ticket, creating a revision history. Each revision includes the date and time of the modification, the user who made the changes, and the specific fields that were modified.
* Notifications (Process 7.4): When a ticket is modified, the system sends notifications to the relevant parties, such as the assigned team or individual, informing them about the changes made. The notifications include a summary of the modifications and a link to view the updated ticket details.
* Approval Workflow: In some cases, ticket modifications may require approval from the assigned team or a designated approver before they are formally incorporated into the ticket. The system can be configured to include an approval workflow, where the proposed changes are reviewed and either approved or rejected.
* Modification Notifications: In addition to the standard notification workflows, the system can be configured to send specific notifications related to ticket modifications:
  + Modification Alert: The assigned team or individual receives an alert when a ticket they are working on has been modified by the user. This ensures they are aware of any new information or changes that may impact their work on the ticket.
  + Modification Summary: The system can generate a periodic (e.g., daily or weekly) summary of all modifications made to tickets within a specific timeframe. This summary is sent to relevant stakeholders, such as managers or administrators, to keep them informed about the overall ticket modification activity.
  + Modification Reporting: The system provides reporting capabilities to track and analyze ticket modifications. Reports can include metrics such as:
  + Modification Frequency: The number of modifications made to tickets over a given period.
  + Modification Types: The breakdown of modifications based on the fields that were changed (e.g., subject, description, category).
* Modification Auditing: The system maintains a comprehensive audit trail of all ticket modifications. The audit trail includes details such as the user who made the modification, the timestamp, the specific changes made, and any associated justifications or comments. The audit trail ensures accountability, enables tracking of ticket history, and facilitates investigations or troubleshooting if needed.

The ticket modification feature empowers users to provide updated and accurate information throughout the ticket lifecycle. It promotes collaboration and communication between users and the assigned teams, ensuring that all relevant details are captured and considered in the problem-solving process. By enforcing appropriate access controls, validation checks, and notification workflows, the system maintains the integrity and reliability of the ticket data while allowing for necessary modifications.

**Requirement ID: DPSR-RN-012-01-12**

**Process ID: 8.1, 8.2, 8.3, 8.4**

**Ticket Reactivation**

**Operating Specification:**

The system provides a ticket reactivation feature that allows users to reopen previously resolved or closed tickets if the underlying issue persists or if additional assistance is required. This feature ensures that recurring or related problems are effectively tracked and addressed.

* Reactivation Eligibility (Process 8.1): The system defines specific criteria for tickets that are eligible for reactivation. These criteria may include:
  + Ticket Status: Only tickets that have been marked as "Resolved" or "Closed" can be considered for reactivation. Tickets that are still in an active state (e.g., Open, In Progress) cannot be reactivated.
* Reactivation Request (Process 8.2): To initiate a ticket reactivation, the user takes the following steps:
  + Ticket Search: The user searches for the specific ticket they want to reactivate using the system's search functionality. They can search based on ticket ID, subject, or other relevant criteria.
  + Reactivation Button: Once the desired ticket is found, the user clicks on a dedicated "Reactivate" button or link associated with the ticket.
  + Reactivation Reason: The system prompts the user to provide a reason for reactivating the ticket. The user enters a brief explanation of why the ticket needs to be reopened, such as the recurrence of the issue or the need for further assistance.
* Reactivation Workflow (Process 8.3): Upon receiving a ticket reactivation request, the system follows a predefined workflow:
  + Notification: The system sends a notification to the original assignee or the team responsible for the ticket, informing them about the reactivation request. The notification includes the ticket details and the user's reason for reactivation.
  + Assignee Review: The assignee or the responsible team reviews the reactivation request and determines whether it is valid and requires further action. They assess the provided reason and any additional information to make an informed decision.
  + Approval or Rejection: Based on the review, the assignee or the responsible team can either approve or reject the reactivation request. If approved, the ticket is reopened and moves back to an active state (e.g., Open or In Progress). If rejected, the user is notified with an explanation for the rejection.
  + Reactivation Notifications: Upon successful reactivation, the system sends notifications to the user and any other relevant stakeholders, informing them that the ticket has been reopened and is being actively worked on.
* Reactivation Data Preservation (Process 8.4): When a ticket is reactivated, the system preserves all the previous data and history associated with the ticket. This includes:
  + Original Ticket Details: The original ticket submission details, such as the subject, description, category, and priority, are retained.
  + Comments and Updates: All comments, updates, and attachments added to the ticket during its previous lifecycle are preserved and accessible.
  + Resolution Details: If the ticket was previously resolved, the resolution details, including the resolution date, the assignee who resolved it, and any associated notes, are maintained for reference.
  + Reactivation Linking: The system establishes a link between the reactivated ticket and the original ticket. This linking mechanism allows for easy tracking and referencing of related tickets. The reactivated ticket may be assigned a new ticket ID while still maintaining a connection to the original ticket ID.
* Reactivation Reporting: The system provides reporting capabilities to track and analyze ticket reactivations. Reports can include metrics such as:
  + Reactivation Count: The total number of tickets that have been reactivated within a specified time period.
  + Reactivation Reasons: A breakdown of the common reasons provided by users for reactivating tickets, helping identify recurring issues or areas that require attention.
  + Reactivation Resolution Time: The average time taken to resolve reactivated tickets compared to the initial resolution time, providing insights into the efficiency of handling reactivated issues.
  + Reactivation Trends: Analysis of reactivation patterns over time, such as identifying peak reactivation periods or correlating reactivations with specific system or process changes.
* Reactivation Auditing: The system maintains a comprehensive audit trail of all ticket reactivation activities. The audit trail includes details such as the user who initiated the reactivation, the timestamp of the reactivation request, the provided reason, and any subsequent actions taken on the reactivated ticket. The audit trail ensures accountability, helps in understanding the history of reactivated tickets, and facilitates process improvements.

The ticket reactivation feature provides a safety net for users, allowing them to reopen previously resolved tickets if the issue persists or if additional support is needed. By incorporating a structured reactivation workflow, data preservation, and reporting capabilities, the system ensures that recurring problems are effectively tracked, managed, and resolved, leading to improved user satisfaction and service quality.

**Requirement ID: DPSR-RN-013-01-13**

**Process ID: 9.1, 9.2, 9.3, 9.4**

**Advanced Search Functionality**

**Operating Specification:**

The system offers an advanced search functionality that allows users to quickly and easily find specific tickets based on various criteria. The search feature is designed to be intuitive, flexible, and efficient, enabling users to locate relevant tickets promptly.

* Search Interface (Process 9.1): The system provides a dedicated search interface that is easily accessible from the main navigation or dashboard. The search interface includes the following elements:
  + Search Bar: A prominent search bar where users can enter their search queries or keywords.
  + Search Filters: A set of predefined search filters that users can select to narrow down their search results. The filters may include criteria such as ticket status, category, priority, assignee, date range, and custom fields.
  + Search Button: A button to initiate the search based on the entered query and selected filters.
* Search Query (Process 9.2): Users can enter various types of search queries to find tickets:
  + Keywords: Users can search for tickets based on specific keywords or phrases that appear in the ticket subject, description, comments, or any other searchable fields.
  + Ticket ID: Users can directly search for a ticket by entering its unique identifier or ticket number.
  + User Information: Users can search for tickets based on the submitter's name, email address, or any other associated user information.
* Search Filters (Process 9.3): The system provides a range of search filters to help users refine their search results. The available filters may include:
  + Ticket Status: Users can filter tickets based on their current status, such as Open, In Progress, Resolved, or Closed.
  + Ticket Category: Users can filter tickets based on predefined categories or departments, such as IT, HR, Facilities, or Finance.
  + Ticket Priority: Users can filter tickets based on their assigned priority level, such as Low, Medium, High, or Urgent.
  + Assignee: Users can filter tickets based on the assigned team or individual responsible for resolving the ticket.
  + Date Range: Users can specify a date range to retrieve tickets submitted or modified within a specific time period.
  + Custom Fields: If the system supports custom fields for tickets, users can filter tickets based on the values of those fields.
* Search Results (Process 9.4): After executing a search, the system displays the search results in a clear and organized manner:
  + Result List: The search results are presented as a list of tickets that match the search query and selected filters. Each ticket in the list includes key information such as the ticket ID, subject, status, priority, and submission date.
  + Pagination: If the search results exceed a certain threshold, the system automatically paginates the results to improve performance and usability. Users can navigate through the pages of search results using intuitive controls.
  + Sorting: Users can sort the search results based on different criteria, such as ticket ID, submission date, priority, or status. This allows users to quickly find the most relevant tickets.
* Saved Searches: The system allows users to save their frequently used search queries and filters as saved searches. Saved searches can be accessed quickly from the search interface or the user's profile. This feature saves time and effort for users who regularly need to perform the same searches.
* Search Performance: The system optimizes the search functionality for fast and efficient retrieval of tickets. It employs techniques such as indexing, caching, and intelligent query optimization to ensure quick search response times, even with a large volume of tickets in the system.
* Search Suggestions: As users type their search queries, the system can provide real-time search suggestions or autocomplete options based on the entered keywords. The suggestions can be derived from commonly used search terms, ticket subjects, or user information, helping users quickly find relevant tickets and reducing the need for manual typing.
* Search Analytics: The system tracks and analyzes search patterns and user behavior to gain insights into commonly searched topics, frequently used filters, and popular keywords. This data can be used to optimize the search functionality, improve ticket categorization, and identify areas where additional knowledge base articles or FAQs may be helpful.

The advanced search functionality empowers users to quickly find the tickets they need, even within a large volume of data. By offering a user-friendly search interface, flexible search options, and optimized search performance, the system enhances productivity, reduces frustration, and improves the overall user experience. The search feature also provides valuable insights into user behavior and preferences, enabling continuous improvement of the ticketing system.

**Requirement ID: DPSR-RN-014-01-14**

**Process ID: 1.1, 1.2, 1.3, 1.4, 1.5**

**Automatic User Onboarding**

**Operating Specification:**

The system includes an automatic user onboarding feature that streamlines the process of granting system access to new employees. By integrating with the organization's HR database or employee management system, the ticketing system ensures that new hires have timely access to the necessary tools and resources.

* HR System Integration (Process 1.1): The ticketing system establishes a secure integration with the organization's HR database or employee management system. This integration allows for the automatic retrieval of new employee information, such as name, email address, department, and role.
* User Account Creation (Process 1.2): When a new employee is added to the HR system, the ticketing system automatically creates a corresponding user account. The account creation process includes the following steps:
  + Unique Identifier: The system generates a unique identifier for the user account, such as an employee ID or a combination of the employee's name and a sequential number.
  + Email Address: The employee's email address, as provided by the HR system, is associated with the user account. This email address serves as the primary means of communication and authentication within the ticketing system.
  + Default Permissions: Based on the employee's department and role information retrieved from the HR system, the ticketing system assigns default permissions and access rights to the user account. These permissions determine the user's ability to submit tickets, view tickets, and perform other actions within the system.
* Welcome Email (Process 1.3): Upon successful account creation, the system sends a welcome email to the new employee. The welcome email includes the following information:
  + Account Details: The email provides the employee with their unique identifier and instructions on how to access the ticketing system.
  + Temporary Password: If applicable, the email may include a temporary password for the employee to use during their first login. The employee is prompted to change this password upon initial access to ensure security.
  + System Overview: The email provides a brief overview of the ticketing system, explaining its purpose, key features, and benefits. It may also include links to user guides, tutorials, or frequently asked questions to help the employee get started.
* User Profile Setup (Process 1.4): After receiving the welcome email, the employee can access the ticketing system using their provided credentials. Upon first login, the system prompts the employee to complete their user profile:
  + Personal Information: The employee can review and update their personal information, such as name, contact details, and profile picture.
  + Communication Preferences: The employee can specify their preferred communication channels for receiving ticket updates and notifications, such as email, SMS, or push notifications.
  + Password Setup: If a temporary password was provided, the employee is required to change it to a strong, personalized password that meets the organization's security guidelines.
* Access and Permissions Assignment (Process 1.5): The system provides an interface for administrators or managers to review and modify the access rights and permissions assigned to new user accounts. This review process ensures that employees have the appropriate level of access based on their roles and responsibilities.
* Onboarding Notifications: The system sends notifications to relevant stakeholders, such as the employee's manager or the IT support team, informing them about the new user account creation. These notifications help ensure that the necessary support and resources are provided to the new employee.
* Offboarding Process: In addition to onboarding, the system also handles the offboarding process when an employee leaves the organization. The offboarding process includes:
  + HR System Synchronization: When an employee is marked as inactive or terminated in the HR system, the ticketing system automatically synchronizes this information.
  + Access Revocation: The system automatically revokes the user's access to the ticketing system, ensuring that former employees no longer have access to sensitive information or the ability to submit or modify tickets.
  + Data Retention: The system retains the user's ticket history and related data for a specified period, in accordance with the organization's data retention policies and legal requirements.
* Audit Trail: The system maintains a detailed audit trail of all user onboarding and offboarding activities. The audit trail includes information such as:
  + User Account Creation: The date and time of user account creation, along with the source of information (e.g., HR system) and the system-generated unique identifier.
  + Permission Changes: Any modifications made to the user's access rights and permissions, including the date, time, and the administrator who made the changes.
  + User Actions: A log of the user's activities within the ticketing system, including ticket submissions, updates, and resolutions.
  + Offboarding Events: The date and time of user account deactivation or termination, along with the reason (if available) and the system-generated offboarding confirmation.

The automatic user onboarding feature streamlines the process of granting new employees access to the ticketing system, reducing manual effort and ensuring timely access to necessary tools and resources. By integrating with the HR system, the ticketing system maintains accurate and up-to-date user information, enhancing security and efficiency. The onboarding process provides a welcoming and informative experience for new users, while the offboarding process ensures the proper termination of access when employees leave the organization. Overall, the automatic user onboarding feature contributes to a seamless and well-managed user lifecycle within the ticketing system.

**HIGH LEVEL DESIGN**

**PURPOSE OF HIGH-LEVEL DESIGN**

The purpose of this High-Level Design Document (HLDD) is to provide a comprehensive and detailed overview of the system architecture, database structure, process design, and user interface design for the Digitalized Problem-Solving Report. This document aims to serve as a central reference point for the development team, stakeholders, and project managers, enabling a clear understanding of the high-level design decisions, system components, and their interactions. It will guide the development process, ensure consistency across different modules, and facilitate effective communication among team members.

**SYSTEM ARCHITECTURE**

**System Overview**

The Digitalized Problem-Solving Report follows a client-server architecture pattern, separating the frontend web application from the backend server components. The frontend application, built using Angular, will provide an intuitive and responsive user interface for users to interact with the system seamlessly across different devices and screen sizes. The backend server, developed using Spring Boot and Java, will handle the business logic, data processing, and communication with the database. RESTful APIs will be used as the communication protocol between the frontend and backend components, ensuring a loosely coupled and scalable architecture.

**System Environment**

The server side of the DPSR will require relatively low amounts of processing power, as the system is designed for a single purpose and not performing other functions. An Intel i5 processor-based system running at 2 to 3 Ghz based on the 12600K chipset will be sufficient. A minimum of 8 Gigabyte of RAM will also be needed. A fast Ethernet network connection for data transfer will also be required, and not less than 100 Mbit/s. Windows 10 or 11 server software will also be required. The client-side computers will require Microsoft Windows 10 Professional at a minimum, with 11 professional recommended. An Intel based processor running at 1Ghz or greater will be sufficient. A minimum of 6 Gigabyte of RAM is also recommended. The web interface will require an internet browser such as Internet Explorer, Chrome, or Mozilla.

**Deployment Architecture**

The Digitalized Problem-Solving Report will be deployed on a robust and scalable infrastructure to ensure high availability, performance, and fault tolerance. The deployment architecture will consist of the following components:

Load Balancer:

* + A load balancer will be employed to distribute incoming traffic evenly across multiple instances of the application servers.
  + It will ensure optimal resource utilization, improve system responsiveness, and provide failover capabilities in case of server failures.

Application Servers:

* + Multiple instances of the application servers will be deployed to handle the processing load and ensure high availability.
  + Each application server will run the Spring Boot backend application, serving incoming requests and executing business logic.
  + Horizontal scaling techniques, such as adding more application server instances, can be applied to accommodate increased traffic and maintain performance levels.

Database Server:

* + A dedicated database server will host the SQL database, ensuring data persistence, integrity, and retrieval.
  + The database server will be configured for optimal performance, with appropriate hardware resources and tuning parameters.
  + Database replication and failover mechanisms will be implemented to ensure data availability and minimize downtime.

Caching Layer:

* + A distributed caching solution, such as Redis or Memcached, will be incorporated to improve system performance by storing frequently accessed data in memory.
  + Caching will reduce the load on the database and provide faster response times for read-heavy operations.

Content Delivery Network (CDN):

* + A CDN will be utilized to serve static assets, such as images, CSS files, and JavaScript files, from geographically distributed edge servers.
  + CDN caching will minimize latency, improve page load times, and reduce the burden on the application servers.

Monitoring and Logging:

* + Monitoring solutions will be deployed to track system health, performance metrics, and resource utilization in real-time.
  + Centralized logging mechanisms will be implemented to collect and aggregate logs from various system components for troubleshooting and analysis purposes.

**Design Considerations** **and Constraints:**

**Process Model**

This sub-section provides the Context Diagram and the first level Data Flow Diagram (DFD) with the process descriptions.

**Digitalized Problem-Solving Report Context Diagram**

**A diagram of a system

Description automatically generated**

**Digitalized Problem-Solving Report Diagram**

**A diagram of a computer program

Description automatically generated**

**Flowchart Diagram:**

**A diagram of a flowchart

Description automatically generated**

**Process Design:**

**Conventions and Standards** **followed**

The design process incorporated standard Gane-Sarson dataflow diagrams.

**Process Model Diagrams:**

**Process 3 Ticket Creation**

**A diagram of a flowchart

Description automatically generated**

**Process** **Descriptions:**

Process 1: Automatic User Onboarding

Process 1.1: HR System Integration

Process 1.2: User Account Creation

Process 1.3: System Sends Welcome Email

Process 1.4: User Profile Setup

Process 1.5: Access and Permissions Assignment

Process 2: Login

Process 2.1: Access Dashboard

Process 3: Ticket Submission

Process 3.1: User Accesses Web Interface

Process 3.2: User Fills Out Ticket Submission Form

Process 3.3: System Applies Access Control

Process 3.4: User Assigns Priority and Delegates

Process 3.5: System Saves User Progress

Process 3.6: System Manages Concurrent Editing

Process 3.7: User Uploads Evidence

Process 3.8: System Generates Digital Report

Process 3.9: Draft Removal

Process 3.10: System Updates Database

Process 4: Trend Analysis

Process 4.1: System Collects Ticket Data

Process 4.2: System Performs Data Analysis

Process 4.3: System Generates Trend Reports

Process 5: Email Notifications

Process 5.1: System Triggers Notification Events

Process 5.2: System Composes Notification Emails

Process 5.3: System Sends Notification Emails

Process 6: Resolution Tracking

Process 6.1: User Accesses Ticket Details

Process 6.2: System Displays Ticket Progress and Status

Process 6.3: User Views Resolution Details

Process 7: Ticket Modification

Process 7.1: User Accesses Ticket for Modification

Process 7.2: User Modifies Ticket Details

Process 7.3: System Saves Modified Ticket

Process 7.4: System Notifies Relevant Parties

Process 8: Ticket Reactivation

Process 8.1: User Requests Ticket Reactivation

Process 8.2: User Provides Reactivation Reason

Process 8.3: System Reopens Ticket

Process 8.4: System Notifies Relevant Parties

Process 9: Advanced Search

Process 9.1: User Accesses Search Functionality

Process 9.2: User Enters Search Criteria

Process 9.3: System Retrieves Matching Tickets

Process 9.4: System Displays Search Results

**INTERFACE DESIGN**

**User Interface**

The user interface is built to follow a linear sequence. It is also our goal to produce an intuitive yet efficient interface because this system will be used by many employees on a daily basis. The interface will be both graphical and key based. Control commands to perform routine functions will be available for “power users”, as well as buttons for the users that are more comfortable with more of a graphical interface.

**Screen Hierarchy**

The following screen samples show the technique we will employ across the application. Users will become familiar with this type of interface

**Screen 1**: User Log-in

**Screen 2**: The provided image shows the dashboard screen of the DPSR system, which serves as the central hub for users to access and manage various aspects of the ticketing system.

**Screen 3**: The provided image shows the "Create Ticket" screen, which appears after clicking the "Create Ticket" button on the dashboard. This screen allows users to enter the necessary information to create a new ticket in the system.

**Screen 4**: The provided image shows the "Problem Analysis" step within the "Create Ticket" process. This step focuses on gathering detailed information about the problem and conducting a thorough analysis.

**Screen 5**: The provided image shows the final step of the "Create Ticket" process, which focuses on capturing metrics and additional details related to the ticket. This step allows users to provide quantitative information and specify the impact and scope of the problem.

**Screen 6**: The report screen serves as a comprehensive summary of all the information captured during the ticket creation process. It consolidates the details provided by the user across the different steps and presents them in a well-organized and readable format. This screen acts as the final ticket, encapsulating all the relevant information for effective problem resolution and tracking.

This sub-section contains sample layouts of all screens that form part of the system.

**Screen 1:**

**A screenshot of a login screen

Description automatically generated**

**Screen 2:**

**A screenshot of a computer

Description automatically generated**

**Screen 3:**

**A screenshot of a computer screen

Description automatically generated**

**Screen 4:**

**A screenshot of a computer

Description automatically generated**

**Screen 5:**

**A screenshot of a computer

Description automatically generated**

**Screen 6:**

**A screenshot of a report

Description automatically generated**

**Annexure I : Requirements Traceability Matrix**

|  |  |  |
| --- | --- | --- |
| S. No. | Functional Requirement ID | Process ID |
| 1 | DPSR-RN-001-01-01 | 3.2 |
| 2 | DPSR-RN-001-01-01 | 3.9 |
| 3 | DPSR-RN-001-01-01 | 3.10 |
| 4 | DPSR-RN-002-01-02 | 2.1 |
| 5 | DPSR-RN-002-01-02 | 3.3 |
| 6 | DPSR-RN-003-01-03 | 3.4 |
| 7 | DPSR-RN-004-01-04 | 3.5 |
| 8 | DPSR-RN-005-01-05 | 3.6 |
| 9 | DPSR-RN-006-01-06 | 3.7 |
| 10 | DPSR-RN-007-01-07 | 3.8 |
| 11 | DPSR-RN-008-01-08 | 4.1 |
| 12 | DPSR-RN-008-01-08 | 4.2 |
| 13 | DPSR-RN-008-01-08 | 4.3 |
| 14 | DPSR-RN-009-01-09 | 6.1 |
| 15 | DPSR-RN-009-01-09 | 6.2 |
| 16 | DPSR-RN-009-01-09 | 6.3 |
| 17 | DPSR-RN-010-01-10 | 5.1 |
| 18 | DPSR-RN-010-01-10 | 5.2 |
| 19 | DPSR-RN-010-01-10 | 5.3 |
| 20 | DPSR-RN-011-01-11 | 7.1 |
| 21 | DPSR-RN-011-01-11 | 7.2 |
| 22 | DPSR-RN-011-01-11 | 7.3 |
| 23 | DPSR-RN-011-01-11 | 7.4 |
| 24 | DPSR-RN-012-01-12 | 8.1 |
| 25 | DPSR-RN-012-01-12 | 8.2 |
| 26 | DPSR-RN-012-01-12 | 8.3 |
| 27 | DPSR-RN-012-01-12 | 8.4 |
| 28 | DPSR-RN-013-01-13 | 9.1 |
| 29 | DPSR-RN-013-01-13 | 9.2 |
| 30 | DPSR-RN-013-01-13 | 9.3 |
| 31 | DPSR-RN-013-01-13 | 9.4 |
| 32 | DPSR-RN-014-01-14 | 1.1 |
| 33 | DPSR-RN-014-01-14 | 1.2 |
| 34 | DPSR-RN-014-01-14 | 1.3 |
| 35 | DPSR-RN-014-01-14 | 1.4 |
| 36 | DPSR-RN-014-01-14 | 1.5 |

**COMMON CONSIDERATIONS**

**SOFTWARE TEAM**

Our team consists of 7 people: 1 Project Manager, 1 Quality Assurance Engineer, 1 Front End Senior developer, 1 Front End developer, 1 Back End Senior developer, 1 Back End developer and 1 Administrator.

**The Project Manager** : Oversees project development, adheres to timelines, adjusts projects based on feedback, and coordinates with external stakeholders.

**The Programmer** : Including both front-end and back-end developers, they are tasked with code development and documentation, collaborating with the Administrator for high-level design documentation.

**The Quality Assurance Engineer** : Focuses on stress-testing, creating test scenarios to ensure product reliability and identify issues.

**Administrator** : Manages databases and servers, handles access, authentication, and system administration.

**SOFTWARE PROCESS**

**LIFECYCLE BREAKDOWN**

1. **Project Definition Phase**​ : In this phase, we define our problem statement. We understand the purpose of our product. We understand the scope of our project and the relevant limitations of the application.
2. **Project Planning Phase**: This is the phase where we breakdown our project into several smaller tasks which serves as a milestone in establishing a robust way to solve our problem statement. In this phase we make an estimate of the general timeline regarding our project activities. We also estimate the resources and cost required to be spent for this project.
3. **Project Design Phase**: This is one of the most crucial phases of the software engineering process model. In this phase, we design the blueprint of our application. We focus on the functional requirements of our project and develop use cases to better understand how we want our project to be implemented in real case scenarios. We also designed a flow chart to better understand the flow of information and data in our project.
4. **Project Production Phase**: The code will be designed according to this high-level document. To make sure the code, updates, and documentation are efficient and comprehensible, a number of readability and complexity measures will be employed. Codes will be made reusable whenever feasible.
5. **Project Implementation Phase**​: This is the phase where we decide on the appropriate technology to be used for the implementation of the blueprint designed in step 3. We dedicate this phase in developing the core functionalities and models of our software project as in web server development, Client UI and much more.
6. **Project Maintenance Phase**: This is also one of the important phases of our project. Here, some of the team members develop programs that act as testers that test the functionality of the product and determine whether the model is giving us the appropriate results or not. In the first phase we check each module of the product independently called “Unit testing”. In the second phase, we check the effects of each module on the other in terms of their functioning, also called Integration testing.

**SOFTWARE PRODUCT DESCRIPTION**

**Enhanced Efficiency**: Automate the entire ticketing process, reducing the time required for problem reporting, analysis, and resolution. Experience a significant boost in operational efficiency as our software handles repetitive tasks with precision.

**Improved Accuracy**: Minimize human errors associated with manual ticket creation and processing. Our digitalized system employs advanced validation checks and standardized procedures to ensure accurate information capture and processing.

**Increased Transparency**: Provide stakeholders with real-time visibility into the status of reported issues. Automated status updates, notifications, and a user-friendly interface enhance transparency and communication, keeping everyone informed throughout the resolution process.

**User-Friendly Interface**: Designed with both ticket submitters and support staff in mind, our software boasts an intuitive and user-friendly interface. Accessibility is key, catering to users with varying levels of technical expertise and ensuring a positive overall user experience.

**Performance:** Our software sets the standard for performance, ensuring fast processing of problem reports and resolutions. With optimized resource utilization, minimal downtime, and seamless scalability, the software adapts effortlessly to the needs of businesses, from startups to enterprises.

**Availability​:** Our software ensures unbroken availability, providing a reliable and accessible solution for continuous problem resolution. Designed for high availability, the software adapts effortlessly to varying workloads, guaranteeing accessibility for users regardless of the operational demands. Elevate your ticketing processes with a solution that prioritizes availability, enabling your team to efficiently address issues without interruption and ensuring a positive user experience.

**SOFTWARE METRICS**

We use different function-oriented metrics like errors per FP, defects per FP, and pages of documentation per FP.

**SOFTWARE EFFORT AND COST**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Hours (Combined)** | **Role(s) Involved** | **Estimated Cost ($)** | **Rates** |  |
| Initial Phase | 120 | Project Manager (60hrs), Senior Developer (30hrs) | $8,700 | Project Manager | $100/hr |
| Planning Phase | 150 | Project Manager (50hrs), Senior Developers (50hrs) | $9,500 | Senior Developers | $90/hr |
| Requirement Collection Phase | 75 | Developers (75hrs) | $5,250 | Developers | $70/hr |
| Analysis Phase | 135 | Administrator (65hrs), Senior Developer (35hrs) | $8,350 | Administrator | $80/hr |
| Design Phase | 165 | Senior Developers (80hrs), Quality Assurance (40hrs) | $9,600 | Quality Assurance | $60/hr |
| Development Phase | 345 | Senior Developers (170hrs), Developers (175hrs) | $27,550 |  |  |
| Testing Phase | 240 | Quality Assurance (240hrs) | $14,400 |  |  |
| Total | 1230 | Total | $83,350 |  |  |

In accordance with the Gantt chart:

Initial Phase

* Tasks: Define Project Purpose, Define Project Scope

Planning Phase

* Tasks: Define Project Requirements, Schedule Tasks

Requirement Collection Phase

* Tasks: Identify assumptions, dependencies & risks, Estimate costs & development time

Analysis Phase

* Tasks: Define use cases, Define flowcharts

Design Phase

* Tasks: Define functional specifications, define system architecture & API Server, Choose technology stack

Development Phase

* Tasks: Develop Web Server, Develop Client UI

Testing Phase

* Tasks: Unit testing, Integration testing, Functional testing, Security testing, Stress testing

**ADDITIONAL REFERENCES**

* Graduate Studies, University of the East, Philippines, Kent Darryl M. Aglibar, K. D., Alegre, G. C., Graduate Studies, University of the East, Philippines, Del Mundo, G., Graduate Studies, University of the East, Philippines, Duro, K. F., Graduate Studies, University of the East, Philippines, Rodelas, N., & Graduate Studies, University of the East, Philippines. (2023). Ticketing system: A descriptive research on the use of ticketing system for project management and issue tracking in it companies. *International Journal of Computing Sciences Research*, *7*, 1066–1075. https://doi.org/10.25147/ijcsr.2017.001.1.90