

User Access Management System

1. Introduction

1.1 Purpose

The purpose of this document is to outline the requirements for a basic User Access Management system. This system allows users to sign up, request access to software applications, and enables managers to approve or reject these requests. The document aims to provide a clear understanding of the system's functionalities, user roles, and how they interact within the system.

1.2 Scope

The system will implement the following features:

- User Registration (Sign-Up)
- User Authentication (Login)
- Software Application Listing and Creation
- Access Request Submission
- Access Request Approval or Rejection

Technologies to be used:

- Java Servlets
- JavaServer Pages (JSP)
- PostgreSQL Database

2. Overall Description

2.1 Product Perspective

The User Access Management System is a web-based application designed to streamline the process of managing user access to various software applications within an organization. It is an internal tool meant to enhance security and efficiency.

2.2 Product Functions

- **User Registration:** New users can sign up and create an account.
- **User Authentication:** Registered users can log in to the system.
- **Software Management:** Admins can add new software applications to the system.

- **Access Request Submission:** Employees can request access to software applications.
- **Access Request Approval:** Managers can approve or reject access requests.

2.3 User Classes and Characteristics

The system has three primary user roles:

1. Employee

- Can sign up and create an account.
- Can log in to the system.
- Can request access to software applications.
- Cannot approve or reject access requests.
- Cannot create new software applications.

2. Manager

- Can log in to the system.
- Can view pending access requests.
- Can approve or reject access requests.
- Cannot request access to software applications.
- Cannot create new software applications.

3. Admin

- Can log in to the system.
- Has all the privileges of an Employee and Manager.
- Can create new software applications.
- Can manage system settings.

2.4 Operating Environment

- **Server-Side:** Java Servlet Container (e.g., Apache Tomcat)
- **Client-Side:** Web Browser with HTML, CSS, JavaScript support
- **Database:** PostgreSQL

2.5 Design and Implementation Constraints

- The system must be developed using Java Servlets and JSP.
- PostgreSQL must be used for database management.
- The project should be managed using Maven.

3. Specific Requirements

3.1 Sign-Up System (SignUpServlet)

Description

Allows new users to register for the system by creating an account with a default role of "Employee."

Functional Requirements

- **Sign-Up Page (signup.jsp)**
 - Fields:
 - Username (Text input)
 - Password (Password input)
 - Role (Hidden field, defaulted to "Employee")
 - Actions:
 - User enters username and password.
 - Upon submission, data is sent to `SignUpServlet`.
- **SignUpServlet.java**
 - Stores the new user in the `users` table with a default role of "Employee."
 - Redirects to the Login Page upon successful registration.

3.2 Login System (LoginServlet)

Description

Allows registered users (Employees, Managers, Admins) to log in to the system.

Functional Requirements

- **Login Page (login.jsp)**
 - Fields:
 - Username (Text input)
 - Password (Password input)
 - Actions:
 - User enters credentials.
 - Upon submission, data is sent to `LoginServlet`.
- **LoginServlet.java**
 - Validates credentials against the `users` table.
 - Manages user sessions.
 - Redirects users based on their role:
 - **Employee**: Redirect to Access Request Page (`requestAccess.jsp`).
 - **Manager**: Redirect to Pending Requests Page (`pendingRequests.jsp`).
 - **Admin**: Redirect to Software Creation Page (`createSoftware.jsp`).

3.3 Software Management (Admin Only)

Description

Admins can add new software applications to the system.

Functional Requirements

- **Software Creation Page (createSoftware.jsp)**
 - Accessible only by Admins.
 - Fields:
 - Software Name (Text input)
 - Description (Text area)
 - Access Levels (Checkboxes: Read, Write, Admin)
 - Actions:
 - Admin enters software details.
 - Upon submission, data is sent to `SoftwareServlet`.
- **SoftwareServlet.java**
 - Adds new software into the `software` table.

3.4 Access Request System (Employee)

Description

Employees can request access to software applications.

Functional Requirements

- **Access Request Page (requestAccess.jsp)**
 - Accessible only by Employees.
 - Fields:
 - Software Name (Dropdown, dynamically populated)
 - Access Type (Dropdown: Read, Write, Admin)
 - Reason for Request (Text area)
 - Actions:
 - Employee selects software and access type, provides a reason.
 - Upon submission, data is sent to `RequestServlet`.
- **RequestServlet.java**
 - Stores the request in the `requests` table with a status of "Pending."

3.5 Approval System (Manager)

Description

Managers can approve or reject access requests.

Functional Requirements

- **Access Approval Page (pendingRequests.jsp)**

- Accessible only by Managers.
- Displays a list of pending requests, including:
 - Employee Name
 - Software Name
 - Access Type
 - Reason for Request
- Actions:
 - Manager can approve or reject each request.
 - Actions are sent to `ApprovalServlet`.
- **ApprovalServlet.java**
 - Updates the status of the request in the `requests` table to "Approved" or "Rejected."

4. Data Requirements

4.1 Database Design (PostgreSQL)

Tables to Create

1. **users**
 - **Columns:**
 - `id` (Primary Key, Auto-Increment)
 - `username` (Text, Unique)
 - `password` (Text)
 - `role` (Text: Employee, Manager, Admin)
2. **software**
 - **Columns:**
 - `id` (Primary Key, Auto-Increment)
 - `name` (Text)
 - `description` (Text)
 - `access_levels` (Text: Read, Write, Admin)
3. **requests**
 - **Columns:**
 - `id` (Primary Key, Auto-Increment)
 - `user_id` (Foreign Key referencing `users`)
 - `software_id` (Foreign Key referencing `software`)
 - `access_type` (Text: Read, Write, Admin)
 - `reason` (Text)
 - `status` (Text: Pending, Approved, Rejected)

5. Deliverables

- **Source Code:**
 - Java Servlets:
 - `SignUpServlet.java`
 - `LoginServlet.java`
 - `SoftwareServlet.java`
 - `RequestServlet.java`
 - `ApprovalServlet.java`
 - JSP Pages:
 - `signup.jsp`
 - `login.jsp`
 - `createSoftware.jsp`
 - `requestAccess.jsp`
 - `pendingRequests.jsp`
- **Database Scripts:**
 - PostgreSQL script to create `users`, `software`, and `requests` tables.
- **Documentation:**
 - README file with setup and run instructions.

6. Evaluation Criteria

- **Basic Functionality:** The system should allow users to sign up, log in, create software (Admin), and request access (Employee).
- **Code Structure:** Servlets, JSP files, and database interactions should be well-organized.
- **Database Interaction:** Requests and approvals should be correctly handled in PostgreSQL.
- **Completeness:** All required features should be working as described.

7. Roles Summary

The system includes **three** distinct user roles:

1. **Employee**
 - Primary user who requests access to software.
 - Default role upon sign-up.
2. **Manager**
 - Oversees access requests and makes approval decisions.
 - Does not have the ability to create software applications.
3. **Admin**
 - Manages the creation of software applications.
 - Has full access, including all Employee and Manager functionalities.

Understanding these roles is crucial, as they define the permissions and access levels within the system. Each role has specific capabilities that align with their responsibilities in the organization's access management process.