

Lead Developer: Brijesh Nishad Roll No: 2101321550026 College: GNIOT - Greater Noida Institute of Technology

Contents

1	Introduction	2
2	Project Objectives	2
3	Features	2
4	Project Structure	2
5	Files	4
6	Database Design	5
6.1	Tables	5
7	Data Flow and Interaction	5
8	Security Measures	6
9	Conclusion	7
10	References	7

1 Introduction

The Sub-Domain Management System is a comprehensive web application designed to facilitate the management of subdomains for users. The system provides a user-friendly interface for customers to register, log in, manage subdomains, and seek customer support. This project aims to automate and streamline the process of subdomain creation, activation, and management while ensuring data security and integrity.

2 Project Objectives

The primary objectives of the project are:

- To create an intuitive and user-friendly interface for customers to manage their subdomains.
- To automate the process of subdomain creation, activation, and management.
- To ensure the security and integrity of user information and subdomain details.
- To provide efficient customer support for addressing user queries and issues.
- To offer a robust backend infrastructure that handles user requests effectively and efficiently.

3 Features

The Sub-Domain Management System includes a variety of features designed to enhance user experience and operational efficiency:

- Customer Registration and Login: Allows users to create accounts and log in securely.
- Dashboard Overview: Provides users with a centralized view of their subdomains and account information.
- Subdomain Status Checking: Enables users to check the status of their subdomains.
- Subdomain Registration: Facilitates the creation of new subdomains.
- Control Panel for Users: Provides tools for managing subdomains, including activation and deactivation.
- Subdomain Activation: Handles the activation process for subdomains.
- Customer Support: Offers support services to assist users with their queries and issues.

4 Project Structure

The project is structured into several directories and files, each serving a specific purpose. The detailed structure is as follows:

- **src**: Contains Java source files, including Servlets, models, utilities, and DAO classes.

- **Servlets:** Java classes that handle HTTP requests and responses. Key servlets include:
 - * **ActivateSubdomainServlet.java:** Manages subdomain activation.
 - * **RegistrationServlet.java:** Handles user registration.
 - * **LoginServlet.java:** Manages user login.
 - * **SubdomainStatusServlet.java:** Provides subdomain status updates.
 - * **SupportServlet.java:** Handles customer support queries.
- **Models:** Java classes that represent the data structures used in the application. These include:
 - * **User.java:** Represents user data.
 - * **Subdomain.java:** Represents subdomain data.
- **Utilities:** Helper classes that provide common functionalities such as:
 - * **DatabaseConnection.java:** Manages database connections.
 - * **InputValidator.java:** Validates user inputs.
- **DAO (Data Access Object):** Classes that handle database operations. Key DAOs include:
 - * **UserDAO.java:** Manages user-related database operations.
 - * **SubdomainDAO.java:** Manages subdomain-related database operations.
- **web:** Contains web content, including JSP files, CSS, JavaScript, and images.
 - **JSP files:** Dynamic web pages generated by combining Java and HTML. Key JSP files include:
 - * **index.jsp:** Landing page for customer registration and login.
 - * **register.jsp:** Registration form for new users.
 - * **login.jsp:** Login form for existing users.
 - * **dashboard.jsp:** User dashboard displaying subdomain statuses and control options.
 - * **createSubdomain.jsp:** Form for creating new subdomains.
 - * **controlPanel.jsp:** Control panel for managing subdomains and accessing support.
 - **CSS:** Stylesheets that define the visual appearance of the web pages.
 - **JavaScript:** Client-side scripts that enhance the interactivity and functionality of web pages.
 - **Images:** Graphical assets used in the web interface.
- **WEB-INF:** Contains configuration files and compiled classes necessary for the web application's deployment.
 - **web.xml:** The deployment descriptor that configures servlet mappings, context parameters, and other settings.
 - **Classes:** Compiled Java classes used by the application, including servlets, models, utilities, and DAO classes.
- **Database:** Contains database scripts and design documents.

- **Database Scripts:** SQL scripts for creating and populating database tables, managing schema changes, and performing other database operations.
- **Design Documents:** Documentation detailing the database schema, table relationships, and other design considerations.

5 Files

The key files in the project include:

- **index.jsp:** Landing page for customer registration and login.
- **register.jsp:** Page for customer registration.
- **login.jsp:** Page for customer login.
- **dashboard.jsp:** Dashboard displaying subdomain status and control panel options.
- **createSubdomain.jsp:** Form for creating a new subdomain.
- **controlPanel.jsp:** Control panel for subdomain activation and customer support.
- **Servlets:**
 - **ActivateSubdomainServlet.java:** Handles the activation of subdomains.
 - **RegistrationServlet.java:** Manages the registration process.
 - **LoginServlet.java:** Manages user login and session management.
 - **SubdomainStatusServlet.java:** Provides updates on subdomain status.
 - **SupportServlet.java:** Handles customer support queries.
- **Models:**
 - **User.java:** Represents the user entity with attributes such as name, email, phone, address, company/team name, username, password, and registration status.
 - **Subdomain.java:** Represents the subdomain entity with attributes such as subdomain name, database name, database username, database password, and creation status.
- **Utilities:**
 - **DatabaseConnection.java:** Provides methods for establishing and managing database connections.
 - **InputValidator.java:** Provides methods for validating user inputs to ensure data integrity and security.
- **DAO (Data Access Object):**
 - **UserDAO.java:** Contains methods for performing CRUD operations on user data in the database.
 - **SubdomainDAO.java:** Contains methods for performing CRUD operations on subdomain data in the database.

6 Database Design

The database design is a crucial component of the Sub-Domain Management System. It consists of two main tables, each with specific attributes designed to store user and subdomain information securely and efficiently.

6.1 Tables

- **User:** This table stores user information. The attributes include:
 - **userID:** A unique identifier for each user (Primary Key).
 - **name:** The name of the user.
 - **email:** The user's email address, used for communication and login.
 - **phone:** The user's phone number for contact purposes.
 - **address:** The residential or office address of the user.
 - **companyName:** The name of the company or team the user is associated with.
 - **username:** A unique username chosen by the user for logging in.
 - **password:** The user's password, stored securely using hashing techniques.
 - **registrationStatus:** Indicates whether the user's registration is complete and verified.
- **Subdomain:** This table stores subdomain details. The attributes include:
 - **subdomainID:** A unique identifier for each subdomain (Primary Key).
 - **subdomainName:** The name of the subdomain.
 - **databaseName:** The name of the associated database.
 - **databaseUsername:** The username for accessing the database.
 - **databasePassword:** The password for accessing the database, stored securely.
 - **creationStatus:** Indicates whether the subdomain has been successfully created and is active.
 - **userID:** A foreign key linking the subdomain to its respective user.

7 Data Flow and Interaction

The data flow within the Sub-Domain Management System follows a systematic process to ensure smooth operation and interaction between different components. Below is a detailed description of the data flow:

- **User Registration:**
 1. A user accesses *register.jsp* and fills in their details.
 2. *RegistrationServlet.java* processes the input data, validates it using *InputValidator.java*, and then calls *UserDAO.java* to store the user information in the database.

- **User Login:**
 1. The user accesses *login.jsp* and enters their username and password.
 2. *LoginServlet.java* validates the credentials by querying the database via *UserDAO.java*.
 3. Upon successful authentication, the user is redirected to *dashboard.jsp*.
- **Subdomain Creation:**
 1. From *dashboard.jsp*, the user navigates to *createSubdomain.jsp* and submits the subdomain details.
 2. *CreateSubdomainServlet.java* processes the request, validates the data, and calls *SubdomainDAO.java* to insert the subdomain details into the database.
- **Subdomain Activation:**
 1. The user activates a subdomain via *controlPanel.jsp*.
 2. *ActivateSubdomainServlet.java* updates the subdomain's status in the database through *SubdomainDAO.java*.
- **Subdomain Status Checking:**
 1. The user checks the status of their subdomains on *dashboard.jsp*.
 2. *SubdomainStatusServlet.java* retrieves the subdomain status from the database using *SubdomainDAO.java* and displays it on the dashboard.
- **Customer Support:**
 1. Users submit their queries via *controlPanel.jsp*.
 2. *SupportServlet.java* processes the queries and stores them in the database for the support team to address.

8 Security Measures

Security is a critical aspect of the Sub-Domain Management System. The following measures are implemented to ensure the protection of user data and system integrity:

- **Password Hashing:** User passwords are hashed using secure hashing algorithms before storage in the database.
- **Input Validation:** All user inputs are validated to prevent SQL injection and cross-site scripting (XSS) attacks.
- **Secure Database Connections:** Database connections are established using secure protocols and are managed efficiently to prevent unauthorized access.
- **Session Management:** User sessions are managed securely to prevent session hijacking and fixation.
- **Access Control:** Proper access control mechanisms are in place to ensure that users can only access their own data and functionalities.

9 Conclusion

The Sub-Domain Management System provides an efficient and user-friendly solution for managing subdomains. By automating the process of subdomain creation, activation, and management, the system reduces administrative overhead and enhances user satisfaction. The robust backend infrastructure ensures data security and integrity, while the customer support feature addresses user queries and issues effectively.

10 References

- Project GitHub Repository: <https://github.com/yourusername/subdomain-management>
- Frameworks and Libraries: Apache Tomcat, Nginx, Java EE, JSP, Servlets, MySQL
- Security Best Practices: OWASP Top Ten, NIST Cybersecurity Framework