RAILWAY CROSSING STATUS

Algorithm:

**1)index.jsp**: This is the home page of the application, provides links for user login

and admin login.

2) **login.jsp**: It contains the login form for users. It collects the username and

password from the user.

3) **Login.java** (servlet): Is responsible for handling the user login process and

retrieves the username and password from the request parameters, performs a

database query to validate the credentials, and redirects the user to the

"userhome.jsp" page if the login is successful. Otherwise, it displays an error

message.

4) **Registration.jsp**: Here registration form for users collects the username, password, and email from the user.

5) **Register.java** (servlet):Servlet handles user registration process. It retrieves

the username, password, and email from the request parameters, creates aobject, and uses a Register-Daoto insert the user into the database.

6) **RegisterDao.java**: This class is responsible for database operations of user registration. It has a method **insert** that takes a object as input and inserts the user into the database.

7) **Member.java**: It represents a user and contains their username, password, and email.

8) **Userhome.jsp**: This is user's home page. It displays a list of railway crossing information. It also provides links to navigate to other pages like search and favorite crossings.

9. **Search.jsp**: This displays a simple form where users can enter a keyword to

search for railway crossings. It submits the form data to the **searchResult.jsp** page.

10. **SearchResult.jsp**: Processes of search query entered by the user and performs a db query to retrieve matching railway crossings. It displays the results in a table format.

11. **Favorite.jsp**: To display the user's favorite railway crossings. here the code provided is similar to the **SearchResult.jsp** file, so it appears to be a duplication or a mistake.

12. **AddToFavorite.java (Servlet):** This is responsible for adding to the user's favorites. It establishes a db connection, prepares an SQL statement to insert the favorite item, executes the statement, and then redirects the user back to the user home page.

13. **RemoveFromFavorite.java (Servlet):** It looks like there is an error in copying the

code, as the file is identical to AddToFavorite.java.Based on the name, it is likely intended to remove an item from the user's favorites. You would need to

provide the correct code for this functionality.

14. **Adminlogin.jsp:** Jsp file is a view that displays search results from the adminhome table. It sets up a data source and performs a query to search for records that match a given keyword in the Name or Address columns.

15. **AdminLogin.java (Servlet):** This servlet handles the login functionality for the admin

user. It receives the username and password parameters, establishes a connection, and executes a query to check if the provided credentials are valid.

If the login is successful, it forwards the request to the adminhome.jsp page.

Otherwise, it displays an error message indicating a failed login attempt.

16. **admin-home.jsp**: This JSP file displays search results for railway crossings.

17. **Add-Rail.jsp**: This file contains a form for adding a new railway crossing with input fields as the name, address, landmark, train schedule, person in charge, and status of the crossing.

18. **RailCross.java**: This class defines a **RailCross** object that represents a railway crossing. It has private fields for the crossing's attributes and corresponding getters and setters.

19. **RailCrossing.java**: This servlet receives form data from **Addrail.jsp** and inserts a new

railway crossing into the MySQL database using the **RailCross** object.

20. **Update.jsp**: This JSP file displays a form for updating an existing railway crossing. It is

similar to **Addrail.jsp** but pre-fills the form with the existing crossing's data.

21. **UpdateProcess.jsp**: Handles the update process by updating the corresponding record in the MySQL database based on the form data received from **Update.jsp**.

22. **Delete.jsp**: This JSP file handles the deletion of a railway crossing record from the

1. MySQL database