**Railway Crossing Status**

**Project Agenda:** Build an application to communicate the status of railway crossings in advance

**Scenario:**

James Tech Pvt. Ltd. hired you as a Full Stack Developer. They intend to develop an application that can communicate the status of a railway crossing well to the public in advance. You are asked to develop a prototype of the application. The application prototype will then be presented to the relevant stakeholders for budget approval. Your manager has set up a meeting where you are asked to present the following in the next 15 working days (2 weeks):

* Specification document: application capabilities, appearance, and user interactions
* Java EE concepts to be used in the project
* Frontend and backend technologies to be used
* Database to be used

**Tasks you will perform will include:**

* Develop the application using Java EE
* Work with Java Server Page to create a frontend
* Implement Servlet to develop a backend
* Perform CRUD operations using JDBC or Hibernate for MySQL

**Operations to be implemented in the application:**

* Features and operations for the public:
* Create an Account: Registration and login operations for public users
* Fetch the details of the railway crossings and display them on the frontend for the public
* Display the status of the railway crossing: open or closed
* Option to search railway crossings
* Option to mark a railway crossing as favorite
* View that list of Favourite Railway Crossings
* Government level operations:
* Login to a specified admin account to access a dashboard
* Add a railway crossing to the application
* Delete a railway crossing from the application
* Search for a specific railway crossing
* Update the status of a railway crossing: open and close
* Navigation option to access all features with ease

**Flow and features of the application:**

* Document the flow of the application and prepare a flow chart
* List the core technologies and algorithms being used to complete this application
* Code to display the welcome screen. It should display:
* Application name and the developer details
* The details of the user interface such as options displaying the user interaction information
* An option asking to navigate to either the public or government sections
* The first option is a separate webpage for the public:
* It must provide an option to authenticate the user. The user can create an account with basic details like name, email, password, and login if the account exists.
* It should fetch the details of railway crossings.
* With each railway crossing listed, it must display the status of the railway crossing
* An option to search should be implemented to search for a railway crossing by name, like how we search on Google.
* User can mark a railway crossing as a favorite.
* You must display the list of favorite railway crossings separately.
* The second option is used by the Railway Department to manage railway crossings. It should include:
* A form through which an administrator can navigate to the government dashboard using a pre-created email and password in the database.
* Add a railway crossing to the application:
* You can create a form in HTML that can send an HTTP POST request to the Servlet to add the details to the database. Remember to use JDBC or Hibernate as a backend framework.
* Consider the Railway Crossing Model to have the following attributes:

1. Name
2. Address (can have latitude and longitude as well)
3. Landmark
4. Train schedules (probably the timings as a textual description)
5. Details of the person in charge of the railway crossing
6. Status

* Delete a railway crossing from the application
* Display a list of railway crossings with an option to delete them within the Table **View**
* Return a suitable message upon successful deletion of the railway crossing
* Option to update the status of the railway crossing
* It can have the default state as open
* The state can be updated to closed when a train is approaching the crossing
* A proper navigation bar can be created using HTML5 to facilitate easier access to the mentioned options.
* Exceptions should be handled for invalid input
* Implement the appropriate concepts such as exception handling, collections, and searching techniques for source code optimization and increased performance

**Requirements to be met:**

Please ensure you have accomplished the following by the end of your application completion:

* End user authentication to access the application
* Listing of railway crossing sites with status
* Favourite railway crossings for end users
* Government user authentication to access admin Dashboard
* CRUD operations for Railway Crossing Management
* Update Status of railway crossing site

**Screenshots:**

**User end**

Task 1: Create an account or Register

Graphical user interface, application

Description automatically generated

Task 2: Log in

Graphical user interface, application

Description automatically generated

Task 3: Display Railway crossing data

**Graphical user interface, application

Description automatically generated**

Task 4: Railway crossing status

**Graphical user interface, text, application

Description automatically generated**

Task 5: Search Railway Crossing

**Graphical user interface, application

Description automatically generated**

Task 6: Mark railway crossing as Favourite

**Graphical user interface, text, application

Description automatically generated**

**Admin-End:**

Task 7: Add Crossing

**Graphical user interface, text, application

Description automatically generated**

Task 8: Manage Crossings

Graphical user interface

Description automatically generated with medium confidence

Task 9: Delete a Crossing

**Table

Description automatically generated with medium confidence**

Task 10: Update Crossing details

**Graphical user interface, application

Description automatically generated**