Documentation for Monster Sighting

Outline

1. Overview
2. Requirements
3. Technologies Used
4. Webpages
5. Home Page
6. Monsters
7. Locations
8. Organizations
9. Sightings
10. Species

Outline

1. Overview

As Halloween approaches, there has been a heightened awareness of monsters in our midst. The frequency of monster sightings is increasing at an alarming rate. Given this development, I have developed a database and data layer for a Spring Boot Web Application. This monster sightings app allows user to input monster information and track their most recent locations.

1. Requirements
2. Back End
3. It must keep track of all monster information.
   1. Monsters have names, descriptions, and a power.
   2. Monsters can be affiliated with one or more organization.
4. It must keep track of all location information:
   1. Locations have names, descriptions, and address information
5. It must keep track of all monster organization information:
   1. Organizations have names, descriptions, and contact information.
   2. Organizations have members.
6. A user must be able to record a monster sighting for a location and date.
7. The system must be able to report all monsters sighted at a particular location.
8. The system must be able to report all locations where a monster has been seen.
9. The system must be able to report all sightings (monster and location) for a date.
10. The system must be able to report all members of an organization.
11. The system must be able to report all organizations a monster belongs to.
12. Front End
13. It must have a screen(s) to create, view, edit, and delete monsters in the system.
14. It must have a screen(s) to create, view, edit, and delete powers in the system.
15. It must have a screen(s) to create, view, edit, and delete locations in the system.
16. It must have a screen(s) to create, view, edit, and delete organizations in the system.
17. It must have a screen(s) to create, view, edit, and delete sighting (monster, location, and date) in the system.
18. It must have a home page that displays general information about the application, navigation to all the other pages, and a newsfeed of the latest 10 sightings in the database
19. Technologies Used
20. This project connects to a MySQL database where the information is stored. Java Persistence API (JPA) is used to communicate with the database.
21. The Spring Boot Web Application uses Spring MVC to handle HTML requests. The Thymeleaf view engine is used for the webpages.
22. Webpages
23. Home Page
24. This page allows the user to navigate to the other web pages. It also displays the 10 most recent sightings in reverse chronological order with the sighting’s ID, date, location, and the name of the monster that was sighted.
25. Monsters
26. This page allows the user to create, view, update, and delete monsters
27. Monsters have a name, description, and a chosen power. Users can also input any organizations to which the monster belongs.
28. The webpage also displays locations at which they have been sighted.
29. Locations
30. This page allows the user to create, view, update, and delete locations in the database. The location object will have a name, description, address, and city.
31. The user can also view monsters that have been spotted at a location.
32. Organizations
33. Allows users to create, view, edit, and delete organizations in the system.
34. Organizations have a name, description, and phone number.
35. Has a webpage that allows users to view members of that organization.
36. Sightings
37. Users can create, view, edit, and delete sightings which consist of a monster, location, and date.
38. Also allows users to view additional sightings for a given date.
39. Powers
40. Users can create, edit, view, and delete powers that are in the database.
41. Users can input powers that the monsters can have.
42. Users cannot delete the ‘None’ power as this is the default power.