Team Screenshot Design plan

Project - Screenshot Capturing System

CMSC495

Evan Troutt
Ronald Diggs
Ravinder Maini
Jorge NavedoCabrera

December 5, 2021

Overview

This document contains an outline of the program design of the Team Screenshot application. The design is subject to change in the spirit of agile development, but the current iteration detailed in this document should give a fairly reliable overview of the design.

Basics and Dependencies

The Screenshot application is written in Java using the Spring Boot framework and running on the Java Runtime Environment 11. It uses the Maven build automation tool to handle the various dependencies. In order to programmatically take screenshots at regular intervals, the application uses the Selenium browser automation tool, with the Chromedriver extension. For sending emails to users with the screenshots, the JavaMail API is used. We decided to go with a NoSQL approach for the database, so the application uses MongoDB.

Modules and classes

The following is an overview of the modules and classes involved and their functions.

class: ScreenshotApplication

Initializes the Spring Boot application.

class: ScreenshotRunnable

Implements Runnable. Using Selenium, periodically opens a Chrome browser instance and takes screenshots of the websites extracted from the database. Uses the SMTP server to send screenshots to users subscribed to their respective websites via email.

class: ServletInitializer

Initializes the Spring Boot Servlet.

class: CustomAuthenticationSuccessHandler

Determines where the application redirects to upon logging in.

module:service

Contains the classes that interface directly with the MongoDB documents, specifically the User, Role and Site documents.

module:resources

Contains the application properties, the log file, the screenshots themselves, CSS files, and HTML templates.

module:properties

Contains getters and setters for the properties values, extracted from the properties file.

module:controller

Contains the classes that map URLs to functions. Each of these functions does some operation like add or remove a user by interfacing with the Services, then returns a View that is used to render an HTML template.

module:config

Various application configuration files.