Phase 3 Report

Screenshot Capturing System

CMSC 495

December 5, 2021

Evan Troutt
Ronald Diggs
Ravinder Miani
Jorge NavedoCabrera

Milestone schedule

Milestone	Developers	Task start date	Completion date goal	Status
Implement screenshot capturing for multiple websites	Evan Troutt	November 16	November 22	Complete
Implement HTML based emails sent to users	Evan Troutt	November 18	November 23	Complete
Implement screen captures email view	Evan Troutt	November 19	November 25	Not started
Implement functionality to capture screenshots at regular intervals	Evan Troutt	November 20	November 24	Complete
Implement add/edit users	Evan Troutt	November 22	November 27	Complete
Implement add/edit websites	Evan Troutt	November 23	November 28	Complete
Implement error/exception logging	Evan Troutt	November 25	December 2	Complete
Implement screenshot history view	Evan Troutt	November 28	December 6	Complete

As you can see, the application is complete, with the exception of the "Implement screen captures email view". The idea was to have an interface where the user could view and customize the emails he would receive from the service. I made the decision to not include this feature because it seemed highly unnecessary to me, and a lot of work considering that I am the only person working on development. The core functionality of the application is there.

Problems

There were no real problems this week, just some bugs and oddities related to how Spring Boot works. I did end up scrapping the idea of having users be able to customize the interval at which their screenshots are taken. I also did away with the feature of allowing users to customize the emails they receive.

Changes to previous documents

Design plan

I almost entirely rewrote this document. The changes are below:

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.

Test Plan and User's Guide

Location in document	Change
1.1.4 Scope	• Test and validate login information • Test and validate creating a new Screenshot • Test and validate updating a Screenshot • Test and validate utilizing the Capture function • Test and validate utilizing the History View
1.2.1 Program Modules	• Login Module

	• Admin Module • Subscribe/ unsubscribe to website • Send emails to users • Log errors/exceptions
1.3 Features to be tested	• Login Module • Admin Module • Subscribe/ unsubscribe to website • Send emails to users • Log errors/exceptions
1.10.1 System Requirements	➢ Window / Mac / Linux computer
1.10.2 Software	▪ **Operating system** • Windows 10, Mac OS, or Linux ▪ **Communication** • Internet connection ▪ **Java Runtime Environment 11** ▪ **Java SE Development Kit 11 which can be download from the following link:** http://www.oracle.com/technetwork/java/javase/downloads/index.html

	▪ **Maven** ▪ **MongoDB** **Chrome web browser**
1.11 Instructions and Procedures	○ **Instructions and Procedures** ➢ ➢ **1.11.1 How to Set up Environment for the Software** ➢ ➢ 1. Install Maven. ➢ 2. Install MongoDB. ➢ 3. Download Chromedriver for Selenium. Make sure you check the version of Chrome you have installed, and get that same version of Chromedriver. ➢ 4. Add Chromedriver to your PATH, this will vary by what OS you use. ➢ 5. Assuming you're using gmail, google how to set up an application password for it. This is necessary for sending email from the application. ➢ 6. Update src/main/resources/application.yml: ➢ admin should be your email address ➢ password can be anything (these are for logging in as an admin in the application) ➢ smtp:address should be your email address(has to be gmail) ➢ smtp:password should be the gmail application password you set up ➢ spring:resources:static-locations

should be: file:/PROJECTPATH/src/main/resources, where PROJECTPATH is the absolute path of the project

- 7. Make a folder named mongo at the top level of the project.
- 8. Make a folder named captures in the src/main/resources directory.
- 9. open command line/terminal, run: mongod --dbpath /PROJECTPATH/mongo
- 10. In another window or tab, run: mvn spring-boot:run
-
-

1.11.2 How to log in to the software

- 1. Point your web browser to localhost:8080.
- 2. Create a new user, using the button.
- 3. Log in using those user credentials.
-
-
-
-

- **Screen Capturing**

-
- 1. After creating a user and subscribing to a site, log out.
-

	➢ 2. Log back in as admin, using the credentials you defined earlier in application.yml. ➢ 3. The application will start automatically opening a browser window and taking screenshots of the subscribed sites. It will do this every 1 minute.

Evaluation

My original idea for the project was something involving blockchain technology. I think this would have been a more interesting project than what we went with, which was basically just a typical web application. The objection some of the team members had to my idea was that they weren't familiar with blockchain. However I ended up doing all of the development anyway so this wouldn't have mattered. But it's not that bad because I was able to use this project as a means of getting familiarized with things I need to know for my job.