Product: S.S. Media

Client: Doctor Judith E. Rosenbaum

Critical Design Review Document

Black Bear Analytics

Abdullah Karim | Colleen DeMaris | Griffin Fluet | James West | Ryan Handlon

December 17th, 2020

Revision History

Version Number	Release Date	Description
Version 1.0	12/17/2020	Original Release

Acknowledgements

Black Bear Analytics would like to thank Doctor Judith E. Rosenbaum for giving us the opportunity to work on this project. She has been a pleasure to work with on this project and we are excitedly waiting for the future comings of the S.S Media Scraper. We would also like to extend our thanks to Professor Terry Yoo for helping us connect to these wonderful clients.

Table of Contents

List of Figures

List of Tables

1. Executive Summary

Black Bear Analytics Client, Dr. Judith E. Rosenbaum, is an Associate Professor and Chair of the Department of Communication and Journalism. As such, she has a need to easily scrape and compile social media posts off of the internet for her research. Black Bear Analytics was chosen to solve this problem. After careful analysis of the requirements following design standards and methodologies, Black Bear Analytics has designed the S. S. Media Social Scraper. It is a web based application that allows users to log into a website, select specific search criteria, and scrape posts off of various Social Media sites such as Twitter and Instagram. Together, we feel this design meets all the needs and requirements of the client and can feasibly be implemented given the provided time constraints and funding. Details about methodologies, design, testing practices, and more are provided further along in this Critical Design Review Document.

2. Preface

The need for a social media scraping platform came about because of our client's, Dr. Judith E. Rosenbaum, research. A lot of Dr. Rosenbaum and her graduate students' research involves the analysis of social media posts. This includes notable examples such as #boycottnike and research into the acknowledgement of danger in posts at national parks. The hope for this project is to provide a tool that allows for easy gathering of social media posts to make this research easier.

3. Summary

The purpose of this Critical Design Review Document is to provide insight into design of the S. S. Media Social Scraper made by the Black Bear Analytics capstone team commissioned by Doctor Judith Rosenbaum. The S. S. Media shall be a web application able to scrape social media post information from various Social Media platforms including Twitter and Instagram. The document's notable information is split into 10 sections. These are titled Introduction, Purpose, Method, Design, Test, Conclusions, Recommendations, and three Major Appendices. The Introduction will introduce the user to the project, the background of it, and it's pitch. This is followed by the Purpose section, outlining the importance of the project and why it is needed. The Method section outlines how Black Bear Analytics approached the project and it's design, as well as the equipment required for the project. The Design section begins to cover some of the meat of the project. It presents the overall design process of the S. S. Media Social Scraper including areas such as parameters and constraints, approaches to the design, issues encountered process, and specific details. The Test section will cover different areas such as test test conditions, test procedures, and test results. The conclusions section is the wrap up of the overview on the project. It covers a conclusion of the discussion of the design, findings, and a visualization of the design. Next is the Recommendations, an overview of what Black Bear Analytics recommends for the future of the project. The last three Important Sections are the three big Appendices: The Software

Requirements Specification Document, the System Design Document, and the User Interface Design Document. These appendices will provide more detail into their respective areas and the specifics of the design.

4. Introduction

The current systems allowing users to gather information from online social media sites are heavily fragmented and hard for individuals who aren't technically literate to use due to the requirement of learning how to program. Specifically, the UMaine Communications and Journalism department experiences this issue when it comes to gathering information for their research.

5. Purpose

There is a need for a centralized location where researchers of different fields who don't necessarily know how to program can gather information pertaining to social media and public interests to further their interests for the benefit of their universities, foundations, and society.

6. Methods

The team adopted the Scrum agile methodology to keep track of and organize this project. They moved through a V, generating various reports and documentation to support their research and establish requirements, designs, and testing for this project.

To architect a solution that fits our purpose, we developed a scraping architecture. We have a user interface that users utilize to input search criteria pertaining to the research they'd like to do on social media. This includes hashtags, keywords, locations, etc. The user input is then fed into a URL extractor which goes to the web page that contains a gallery of images that are relevant to the search. This gallery is contained within an HTML tree structure (all web pages have an HTML tree structure) which we use to generate links to all other web pages. The generated web pages contain data related to a post that researchers will use in their studies. The HTML tree structures of posts are converted into Document Object Models (DOM) which are processed into data that is aggregated for the user. This process ends with text data being saved out to a .CSV file and picture information being saved to a folder, with the link between them being a unique ID.

Jira is used as an organizational tool where the team keeps a kanban board with relevant stories. Gitlab/Github is our version control and a remote repository for storing code. It also acts to establish jobs for tests, building, and deploying our product. Selenium is used to test our product through automation. Scrapy Python / BS4 Python / Python is used to develop code and modelize

the DOM's described in the Methods section. React JavaScript is used in our front-end to implement a user interface. Figma is used to mockup our product

7. Design

7.1 Design Parameters

Our customer had requested a straightforward and easy to understand design. The same was said for the layout, so we tried to make it as easy to navigate and use as possible. Aside from the basic search features we will be providing, we also wanted to offer some advanced search features for more intricate scrapes. Our skills for front-end design are still fresh. So we attempted to create a design that was made up of simple components that look clean and will come together nicely.

7.2 Design Approach

The design was approached with both simplicity, intuitiveness, and user friendliness in mind. To ensure simplicity, we created each page with large gaps in between components such as buttons, labels, and drop down menus. The components are made to be simple shapes, some requiring a label. This will make the coding of the components to match the design of the mockup much easier. We've also guaranteed user friendliness by designing popup text boxes to help the users at every point during the use of the program. These text boxes will popup when a user hovers over a label and describes what the component does, and how it is used to interact with the rest of the program.

7.3 Design Problems

We are expecting some problems to arise when making the final design such as the dynamic scaling of the windows. This may prove to be difficult since the formatting of the buttons, labels, drop downs, and such may have some unexpected results. When developing a front-end that is web-based, challenges can emerge as to what browser or device the user is connecting from. For every browser that we plan to support, our code base may require small tweaks to ensure compatibility. This is the same for devices that can have a wide range of unique screen sizes that our user interface will need to scale to.

7.4 Design Details

There a few must haves that the customer insisted on. One being having the date selection available for the basic search. This will allow for users to select certain time frames to scrape from. A login feature was also requested so that only allowed individuals can use our program. Another request of our client was to include a built-in tutorial. For this we have included labels for all of the text fields, buttons, dropdowns, etc. They will also include an option to hover the component to display a popup that explains what should go into the component. Along with the tutorial we will also include a very thorough user manual. In the case of any confusion with an aspect of the program the user manual can be referred back to. The last request of our customer was to ensure

that the design was very intuitive and simple since not everyone using this program will be tech savvy.

8. Testing

8.1 Testing Structure

Overall, we will be testing using tests likely written in Python, by one person in a paired programming setup. One person will write the code that is necessary for the creation of the product, while the other person writes the tests that are necessary for the testing of the product. Any tests that are possible to be automated, will be automated. We want to run both regression and integration tests so that we can be sure that each implemented requirement put into place in the SRS will work together.

8.2 Test Software

Currently, there is no specific definition as to what exact testing software we will be using. The winter break will be used to gain and implement more information. However, there are several possible softwares to use. First, we are planning on using GitLab in order to create pipelines and jobs from our test definitions. We will likely use Selenium to help our tests run, as Selenium provides an automated testing process. We were also looking into Travis CI, which is an integration service that is used to both build and test software.

8.3 Test Procedures

So far, there have been ten tests planned for the product. These tests include an authorization test, a scraping test, a search feature test, a data accuracy test, a licensing test, encryption tests, and compatibility tests. These tests are outlined in the SRS document, but have not been conducted yet as there is no product to conduct them on.

9. Conclusions and Recommendations

The S.S. Media Social Scraper is being made with the intent to make scraping social media easy. It is designed as simply as possible in order to allow anyone to use it and understand how to use it. Doctor Rosenbaum will use this product for her research into social media posts, such as looking at hashtags that people post, locations that they are at, and the images that are posted. The S.S. Medial Social Scraper will be put into a web-based format where it will allow anyone with the authorization to go to the URL, log in, and use the tool. We will continue working on this product during the winter break, completing tasks such as setting up a mock system on AWS and updating the system to include the features, as well as creating a full testing plan. The goal for this product is to create an easy-to-use social media scraper that can switch between scraping different platforms and store the data in a searchable table.

10. Bibliography

Banga, S. (2020). *What is Web Application Architecture? Components, Models, and Types*. Hackr.io. https://hackr.io/blog/web-application-architecture-definition-models-types-and-more.

Fulton, J. (2018, December 12). *Web Architecture 101*. https://engineering.videoblocks.com/web-architecture-101-a3224e126947.

Kerins, I. (2019, July 4). Solution Architecture Part 5: Designing A Well-Optimised Web Scraping Solution. https://blog.scrapinghub.com/solution-architecture-part-5-designing-a-solution-estimating-resource-requirements.

Wikimedia Foundation. (2020, October 29). *Web crawler*. Wikipedia. https://en.wikipedia.org/wiki/Web_crawler.

Other References:
Black Bear Analytics SRS Document

UI Design : https://www.figma.com/file/qc0MoZOfnAJrwfYVoisxcN/BlackBearAnalytics_Mockup?node-id=0%3A1

AWS : https://aws.amazon.com/

SDD Document

UIDD Document

Document A – Software Requirements Specification Document

Product: S.S. Media

Client: Doctor Judith E. Rosenbaum

Black Bear Analytics

Abdullah Karim | Colleen DeMaris | Griffin Fluet | James West | Ryan Handlon

October 27, 2020

Revision History

Version Number	Release Date	Description
Version 1.0	10/24/2020	Original Release

System Requirements Specification

Table of Contents

1. Introduction

The purpose of this section is to introduce the reader to both this document and the system requirements for the project, detailing introductory information that will be useful for the reader to know.

1.1 Purpose of this Document

The purpose of the SRS is to clearly outline the features agreed upon by both our client (Doctor Rosenbaum) and Blackbear Analytics for the S.S. Media scraping tool. This document will cover the scope of the product and detail all requirements of the application. This includes functional and non-functional requirements, user interface design, required deliverables, and open issues.

The intended readership of this document includes Developers, Testers, the Client, and End Users. Developers will review this document to more fully understand the product, what must be developed, and help guide their efforts in its creation. Testers can use this document to help guide them in their testing of the application helping guarantee proper functionality of the product. The Client will read this document to have a greater understanding of the product they commissioned as well as guarantee that their development team understands what they need out of the product. The end user would review this document to better understand the product's functionality and intended purpose.

1.2 References

This document does not include any cited information and does not require any references.

1.3 Purpose of the Product

The client for this product is Associate Professor and Chair of the Department of Communication and Journalism Doctor Judith E. Rosenbaum at the University of Maine in Orono. A great deal of hers and her grad student’s research involves the analysis of social media and how people use it. Two notable examples of this are the Boycott Nike hashtag from late 2018 or current research on the analysis of people's perceived risk when taking pictures at national parks, and whether they acknowledge it. An important aspect of this research is data collection, which requires a way to scrape and compile information based on specific hashtags, times, phrases, or location from social media sites. Doctor Rosenbaum has had tools in the past that would do this data scraping for her, however those tools were complex and eventually became outdated. She now needs a new up to date scraper that is easy for the end user to understand.

1.4 Product Scope

The scope of this project is to provide a web-based tool that can scrape information off of social media websites. The application will have a user-friendly design allowing them to login and scrape information from any supported social media platform. Scrapeable websites shall include Twitter and Instagram exclusively. Should time permit, scraping of Reddit, Facebook, Snapchat, and TikTok will also be implemented in that order. The user will be able to scrape for information based off of hashtags, dates, locations, and key words or phrases. The tool will then compile the information into an easily manageable format and provided to the end user. In the base version of the scraping tool, the application will not provide any analysis or manipulation of information after it has been gathered. As can be seen in Figure 1, the system will provide all the functionality required to manage user accounts, login, select scraping specifications, scrape data, and downloading data. Outside actors of the system will include: Researcher, Administrator, and Unauthorized User, the social media platforms being scraped which include but are not limited to Twitter and Instagram, and the storage location of scraped data.

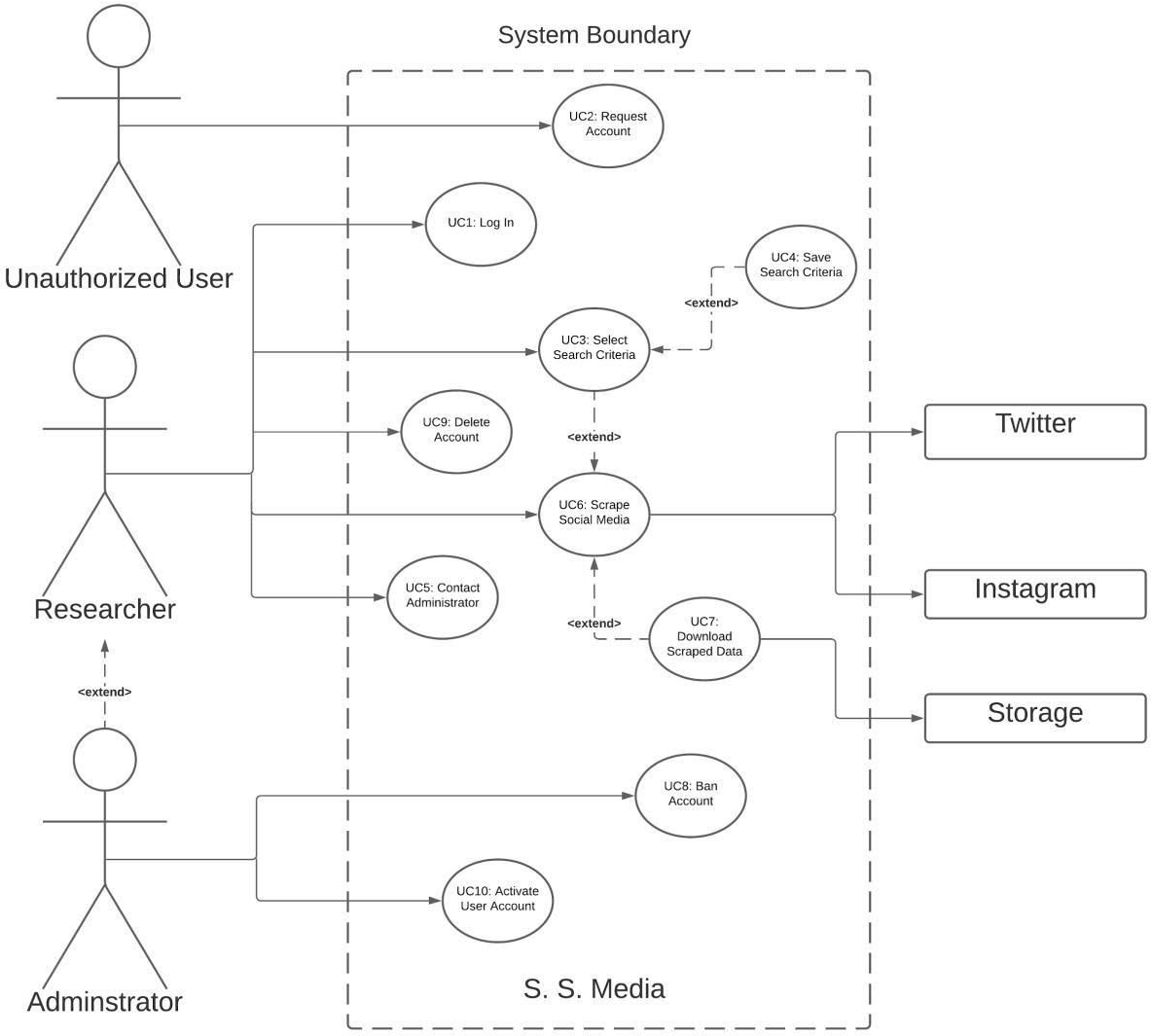


Figure 1: Use Case/Context Diagram of the S.S. Media Application

2. Functional Requirements

This section contains use cases that outline the functionality of our program. Per use case, there is a corresponding acceptance test titled in the use case description. The steps for each use case is written out at the end of the section for Functional Requirements. Functional Requirements are adapted from elicited user stories written while talking with the client about their needs. Use cases for open issues relation to functionality have no corresponding use case but will be included at the time of closing the issue. Post minimum viable product (MVP) functionality is not included in this iteration of the SRS as they're open issues that are undergoing scope definition. For acceptance tests related to each functional requirement, refer to the list of tests in section 3.2 Acceptance Tests.

2.1 Use Case Descriptions

Number	1	
Name	Log In	
Summary	This use case addresses security for our system. Users must log in with a username and password combination that's given to them by an administrator upon request and approval of an account.	
Priority	4	
Preconditions	The user must be on the login screen. They must also have valid credentials.	
Postconditions	Users will be logged into the Home Page which displays their scrape history, settings, and a button that allows for the setup of a scrape.	
Primary Actor	Researcher	
Secondary Actors	Administrator	
Trigger	User enters the login screen	
Main Scenario	**Step**	**Action**
	1	User clicks username field. System activates the name field and allows the user to type into it.
	2	User enters a username. The system saves the information the user has entered.
	3	User clicks the password field. System activates the password field and allows the user to type into it.
	4	User enters password. As the password is being typed, the system displays it as dots to the user while also saving their input for authorization.
	5	System makes the greyed-out login button active and blue so the user can log in.
	6	User clicks the login button. If the user's credentials are valid, they are logged in.
Extensions	**Step**	**Branching Action**
	6a	User failed authentication: Else, red text appears above the username field stating: "Invalid credentials. Please try again or contact the administrator."
Open Issues	Refer to Open Issue 2	

Number	2
Name	Request Account
Summary	Users request an account by clicking a button and filling out/sending a form for administrator approval.
Priority	4

Preconditions	User needs an account	
Postconditions	User submits a form for the user account that's sent to the administrator.	
Primary Actor	Researcher	
Secondary Actors	Administrator	
Trigger	Click the Request Account button from the Login window.	
Main Scenario	**Step**	**Action**
	1	User navigates to the Login window and clicks the request account.
	2	System produces a form with details of Email, First Name, Last Name, and Password text fields.
	3	User fills out the form and clicks the Request Account button.
	4	User is prompted to verify their email and the system sends the verification email with instructions.
	5	User opens the email and clicks the link to the verification page.
	6	System verifies the user.
	7	User is prompted that their account is verified and awaiting approval from an administrator. The system adds the new user account to the Administrator's Approve Accounts list.
Extensions	**Step**	**Branching Action**
	3a	The user does not enter a valid password and is not meeting the systems password criteria. The user is prompted to enter a password with at least 1 uppercase and 1 lowercase letter, 1 special character, 1 number, and be at least 8 characters long.
Open Issues	Refer to Open Issue 2	

Number	3	
Name	Select Search Criteria	
Summary	User selects their search criteria such as hashtags, locations, platform being scraped, and date range.	
Priority	5	
Preconditions	User must be logged in on the home screen.	
Postconditions	Users have a set of search criteria that they can then use to scrape data and/or save the search criteria.	
Primary Actor	Researcher	
Secondary Actors	Administrator	
Trigger	User clicks the New Search button or clicks a saved previous search.	
Main Scenario	**Step**	**Action**
	1	User clicks the New Search button.
	2	User selects the social media platform they are scraping.
	3	System loads search criteria for specified social media.
	4	User inputs basic search criteria into associated boxes. (Hashtags, Locations, Phrases)
Extensions	**Step**	**Branching Action**
	1a	User selects a Saved Search criterion from a list of saved criteria on the Home Page.
	1a1	System loads Scrap Search Criteria page and populates it with saved criteria. Skip steps 2-4.
	3a	User selects Advanced Search

	4a1	User inputs advanced search criteria (Hashtags, Locations, Phrases, Date Range, and/or logic, etc.)
Open Issues	Refer to Open Issue 1	

Number	4	
Name	Save Search Criteria	
Summary	Users can save searched criteria for future use.	
Priority	3	
Preconditions	Users are on the Social Platform Selection screen and have selected search criteria.	
Postconditions	The search criteria of a user is saved with their information.	
Primary Actor	Researcher	
Secondary Actors	Administrator	
Trigger	Save Search Criteria button from the Social Platform Selection screen.	
Main Scenario	**Step**	**Action**
	1	User clicks the Save Search Criteria button.
	2	System creates a popup to prompt for name of the search criteria.
	3	User enters a name in a popup prompt.
	4	System saves that search criteria to the scrape history associated with the user's profile in the user database.
Extensions	**Step**	**Branching Action - N/A**
Open Issues	N/A	

Number	5	
Name	Contact Administrator	
Summary	Users having trouble logging in or who have general issues with the software can click the Contact Administrator button on the login window and submit a form to administrators so issues can be resolved.	
Priority	1	
Preconditions	User is on the Login page.	
Postconditions	User submits a form with the submit button, sending their contact information and a description of the issue to admins.	
Primary Actor	Researcher	
Secondary Actors	Administrator	
Trigger	Click the Contact Administrator Button.	
Main Scenario	**Step**	**Action**
	1	User clicks the Contact Administrator Button.
	2	System provides a form where the user can enter their contact information and description of their issue.
	3	User enters contact information and description of their issues.
	4	User clicks the submit button.
	5	System sends this information to Administrator accounts.
Extensions	**Step**	**Branching Action**
	1a	User clicks the cancel button to retract their contract form.
Open Issues	N/A	

Number	6

Name	Scrape Social Media	
Summary	The system will initiate a search for all relevant information from the selected social media platform.	
Priority	5	
Preconditions	User is on the Social Platform Selection screen and has filled in all relevant search criteria.	
Postconditions	Data relating to the search criteria will be gathered from relevant social media sites.	
Primary Actor	Researcher	
Secondary Actors	Administrator	
Trigger	Click the Perform Scrape button	
Main Scenario	**Step**	**Action**
	1	User clicks the Start Scrape button.
	2	System accesses selected social media platform.
	3	System runs algorithm scraping information with an input of selected criteria.
	4	System finishes running algorithm and prompts user to download scraped data. (see Download Scraped Data Use case).
Extensions	**Step**	**Branching Action**
	4a	User clicks the Halt Scrape button.
	4a1	System stops the scraping algorithm and returns the user to the home screen, throwing out any captured data in the process.
Open Issues	Refer to Open Issue 3	

Number	7	
Name	Download Scraped Data	
Summary	Once a scrape is finished, the data is downloaded to the user's local machine, so they are accessible to the user.	
Priority	4	
Preconditions	User has hit the Scrape button and a scrape has been performed.	
Postconditions	All information associated with a scrape is stored out to excel files and folders on the user's desktop.	
Primary Actor	Researcher	
Secondary Actors	Administrator	
Trigger	Scrape has completed.	
Main Scenario	**Step**	**Action**
	1	The system has met the search criteria's time constraints, and finishes searching.
	2	The system prompts the user if they want to download or discard the scraped data.
	3	User selects download
	4	The system downloads the data to the user's device and saves it to the Downloads folder.
Extensions	**Step**	**Branching Action**
	3a	User selects Discard.
	3a1	The system deletes the scraped data and returns the user to the home page. Skip step 4
Open Issues	Refer to Open Issue 3	

Number	8	
Name	Ban Account	
Summary	Administrators are allowed to ban user accounts that exist.	
Priority	1	
Preconditions	User must be an admin logged in.	
Postconditions	A user account is added to a blacklist.	
Primary Actor	Administrator	
Secondary Actors	N/A	
Trigger	User selects an account from the Edit User Accounts list and clicks the Ban button.	
Main Scenario	**Step**	**Action**
	1	User clicks a checkbox next to a user's name.
	2	User clicks the Ban button.
	3	The system displays a verification popup.
	4	User clicks the Yes, I Am Sure button.
	5	The system removes the account from the user database.
Extensions	**Step**	**Branching Action**
	1a	User clicks the Select All checkbox.
	1a1	User clicks the Ban button.
	1a2	The System returns to the Edit User Accounts page.
Open Issues	N/A	

Number	9	
Name	Delete Account	
Summary	Researchers are allowed to delete user accounts that exist.	
Priority	1	
Preconditions	User must be a logged in researcher. User must be on the settings page.	
Postconditions	A user account is deleted.	
Primary Actor	Researcher	
Secondary Actors	N/A	
Trigger	User clicks the delete account button.	
Main Scenario	**Step**	**Action**
	1	User navigates to the settings page.
	2	User clicks the Delete Account button.
	3	The system displays a verification popup.
	4	User clicks the Yes, I Am Sure button.
	5	The system removes the account from the user database.
Extensions	**Step**	**Branching Action**
	4a	User selects the "No, I am not sure" button.
	4a1	The system returns the settings page.
Open Issues	N/A	

Number	10
Name	Activate User Account
Summary	Administrators must approve user accounts when a form request is received. Approving a user account gives them access to the website beyond the login page.
Priority	2
Preconditions	Must be logged in as an Administrator and on the Edit User Accounts screen.

Postconditions	A user's verified account is activated, giving them access to the tools and functionality.	
Primary Actor	Administrator	
Secondary Actors	Unauthorized User	
Trigger	Account request form is submitted from an Unauthorized User.	
Main Scenario	**Step**	**Action**
	1	Administrator clicks the checkbox next to the new user.
		System displays information about the new user.
	2	Administrator clicks the Approve button.
	3	New user removed from the Approve Accounts list and gains researcher permissions.
Extensions	**Step**	**Branching Action**
	2a	Administrator clicks Deny button.
	2a1	New User is informed they have been denied via email, account is deleted, and removed from the Approve Accounts list.
Open Issues	N/A	

3. Non-Functional Requirements

This section of the document will detail all Non-Functional requirements for this project. All Non-Functional Requirements will be accompanied by a unique id number, a priority, a clear description, and a test that will be used to verify the requirement has been met.

3.1 Non-Functional Requirements

ID	Description	Priority	Tests
1	The system shall authenticate (or prevent authentication) to users within 5 seconds of the user clicking the login button.	5	Authorization Test
2	The system shall save the search criteria successfully 99% of the time when the user clicks the "Save Search Criteria" button.	4	Search Feature Test Scraping Test
3	The system shall return scraped, text-based information in a CSV file 99% of the time when the user clicks the start scrape button.	2	Scraping Test
4	The system shall return scraped, visual-based information 99% of the time when the user clicks the start scrape button	2	Search Feature Test Scraping Test
5	The system shall return scraped information within an overall time period of no more than 1 seconds per post being scraped	3	Scraping Test

6	The system shall load the saved search criteria successfully 99% of the time	5	Search Feature Test Scraping Test
7	The system shall provide accurately scraped data with 98% confidence.	5	Data Accuracy Test
8	The system shall be Open Source.	1	Licensing Test
9	The system shall have 99% of the information in its user database encrypted.	4	Encryption Tests
10	The system shall be compatible with popular platforms including Google Chrome, Firefox, Safari, and Microsoft Edge.	2	Compatibility Test

3.2 Acceptance Tests

Authorization Test
Corresponding Non-Functional Requirement ID's: 1 Corresponding Functional Requirement ID's:

Enter username and password. Start a timer and then click the Login button. When the user is logged in, stop the timer. If the timer is greater than 5 seconds the test fails, otherwise it passes.

Search Feature Test
Corresponding Non-Functional Requirement ID's: 2, 4, 6 Corresponding Functional Requirement Use Cases: 3

Go to a "@socialmediaplatform". Post a "@testdata" on the developer account being used with a keyword to test such as "@parameter" with a unique id that goes with the post so that it's easily identifiable post-scrape. Navigate to the social platform selection screen on the web scraper. Enter selection criteria such as "@parameter". Run the scrape and see if the information provided contains the unique id specified. Repeat per @socialmediaplatform, @testdata, and @parameter. If it succeeds 99% of the time, the test passes. Else, test fails.

Scraping Test
Corresponding Non-Functional Requirement ID's: 2, 3, 4, 5, 6 Corresponding Functional Requirement Use Cases: 6, 7

Navigate to the Social Platform Selection screen on the web scraper. Enter selection criteria. Start a timer and hit the scrape button simultaneously. Keep track of the number of requests. Once the process is done, stop the timer and check for any data produced in the downloads folder. If there is data and the total time taken for the scrape divided by the number of scrapes is under 1 second, the test succeeded. Else, test failed.

Data Accuracy Test
Corresponding Non-Functional Requirement ID's: 7

Scrape information from a social media platform. If there was an image, access the image and see if it's good quality. If it is, use the image's filename as a unique id to check the excel sheet for any corresponding text information. If there is, verify that information and the image correspond to the post on the social media platform that it's describing. If the information is just text (such as a tweet), go to that text's corresponding post on the social media platform and verify that the information is the same. If it succeeds 99% of the time, the test passes. Else, test fails.

Licensing Test
Corresponding Non-Functional Requirement ID's: 8

Open the software licensing agreement. Read it. If the license states the software is Open Source the test passes, otherwise it fails.

Compatibility Test
Corresponding Non-Functional Requirement ID's: 10

Open the web scraper in "@web browser". Run all relevant tests to the application including the Authorization Test, the Search Feature Test, the Scraping Test, the Data Integrity Test, and the testing of basic website functionality. If all tests pass, the compatibility test passes for "@web browser". Repeat test for all applicable browsers.

Encryption Test
Corresponding Non-Functional Requirement ID's: 9 Corresponding Functional Requirement Use Cases: 2, 10

Open the application and create a new account with test user information. On the administrator account, approve the account request. Log in as the new user. Request test user information from the user database. Compare the raw data received to the actual user data. If they do not match, the received data looks encrypted, and isn't in plain text, the test passes. Otherwise the test fails.

Login Test
Corresponding Functional Requirement Use Cases: 1

Get an approved account from through the administrative process. Attempt to log into the platform with fake login credentials. The system will give you an error and tell you to try again. Enter the correct credentials for a user account. The system brings you to the Homepage of the scraper tool within 5 seconds of clicking the Login button. If the system did not give an error or the correct credentials failed to log the user in, this test fails. Else, this test passed.

Save Search Test
Corresponding Functional Requirement Use Cases: 4

Log into the scraper platform with valid credentials. Select the New Search Button. Enter search criteria and a platform for scraping. Click the Save Search Criteria button. Fill out the form that asks for a name of the search and hit submit. Navigate to the Home Page where saved search criteria names are shown and alongside the New Search button. If the search criteria name that was saved is there, click it (Else, the test fails). If the Social Platform Selection screen is brought up and the saved search criteria are auto filled, this test passes. Else, test fails.

Administration Test
Corresponding Functional Requirement Use Cases: 5, 8, 9

Open the application and create a new account with test user information. Log in to the administrator account and approve the new test account request. Log in as the new test user and fill out the Contact Administrator form with sample test information. Go back to the administrator account and check for a notification with the corresponding test data. If it's not there, the test fails, otherwise continue. Ban the new test account. Try to log in to the test account. If the login is successful, the test fails, otherwise continue. On the administrator account, unban the test account. Try to log in to the test account. If login fails, the test fails otherwise continue. On the test account, navigate to the settings page. Click the Delete Account button and select yes on the Are you sure prompt. After being redirected to the login page, try to login to the test account. If the login is successful, the test fails. Else, this test passes.

4. User Interface

See “User Interface Design Document” for *S.S. Media*. This document is a deliverable due on November 22nd, 2020 and will be prototyped, with a final version not available until then.

5. Deliverables

This section of the document provides a list of all deliverable items throughout the course of this project. Each deliverable will be accompanied by information about its due date, what format it is to be delivered in, method of delivery, and any notes.

5.1 Deliverable Schedule

Name of Deliverable	**Format of Deliverable, Method of Delivery**	**Date of Delivery to Client**	**Notes**
Budget Estimate	.xlsx (Excel), through email and GitHub	October 14th, 2020	
Systems Requirement Specification	PDF, through email and GitHub	October 23rd, 2020	At this time Doctor Rosenbaum can look over the document and suggest any changes before signing it.
System Design Document	PDF, through email and GitHub	November 9th, 2020	
User Interface Design Document	PDF, through email and GitHub	November 22nd, 2020	
User Manual	PDF, through email and GitHub	April/May, 2021	
Administrator Manual	PDF, through email and GitHub	April/May, 2021	
Update Manual	PDF, through email and GitHub	April/May, 2021	This manual is used to find places in the code (in addition to code comments) that need to be updated if social media platforms update their APIs so that future developers can still use this program once platforms have changed. It will also outline how to use the program for new users as well as have a changelog of information
Executable Program	.src files, website link through email and GitHub	April/May, 2021	

6. Open Issues

The Open Issues section is a list of all currently known problems or questions pertaining to software requirements that need answering. Each open issue will be accompanied by a brief explanation of the issue, a timeline for when the issue will be resolved, and a specification of who is in charge of solving it.

Open Issue #1: Search Criteria

It is unknown whether the specified search criteria (Hashtags, Locations, Phrases) is viable for all social media platforms being scraped. For example, Twitter uses geolocation tags while Instagram does not all the time. There may also be special search criteria that are specific to individual social media. These search criteria need to be researched and a client meeting about the topics needs to happen. This issue shall be resolved within 20 days of the signing of this document and is assigned to Developer Griffin Fluet.

Open Issue #2: Number of Users

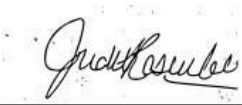
The number of supported users at any given time is yet to be determined. Currently, we go with one user who carries the role of Administrator and Researcher, but we'd like to see if this can be expanded to support more than one user at a time. This criterion needs to be researched and a client meeting needs to happen to resolve the situation. This issue shall be resolved within 14 days of the signing of this document and is assigned to Developer Colleen DeMaris.

Open Issue #3: Size of Files

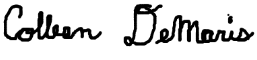
How much information that can be collected in a single scraping session and the potential size of that file are potential problems. Allowing for scrape sessions and file sizes that are too big could result in a download or transfer that could break something. This criterion needs to be researched and consulted on with the client. This issue shall be resolved within 14 days of signing this document and is assigned to Developer James West.

Appendix A – Agreement Between Customer and Contractor

The client (Doctor Rosenbaum) and Blackbear Analytics agree that all requirements in this document will be implemented in good faith. All requirements will be implemented as close to the specifications as possible, with margins for error being slim or none. In the event that implementation must diverge from the specifications, the development team will notify the client as soon as possible and work together with the client until a suitable agreement is reached. If the client would like to add requirements after signing this document, Blackbear Analytics bears no responsibility for the failure to implement new requirements but will attempt to add them in good faith. Should any new requirements be added, the deliverables schedule in section 5 of this document will be updated immediately to reflect changes, and the requirements discussed will be reflected in this document by the next deliverable deadline and changes will be finalized with the client.

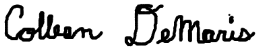
Name	Signature	Date
Customer:		
Judith E. Rosenbaum		10/25/2020
Comments:		

Team:

Ryan M. Handlon		October 24th, 2020
Abdullah I. Karim		October 24th, 2020
Griffin L. Fluet		October 24th, 2020
Colleen DeMaris		October 24th, 2020
James West		October 24th, 2020

Appendix B – Team Review Sign-off

By signing your name below, you acknowledge that you are a member of Blackbear Analytics and have read the document with a solid comprehension of the scribed materials. You agree to complete all requirements stated on this document as is, in good faith. Should any new requirements come up, you agree to assist in re-drafting this document for approval as described in Appendix A. You agree that it is not required to complete requirements added after this version is released, but understand that all requirements, new or old, must be implemented in good faith.

Name	Signature	Date
Ryan M. Handlon **Comments:**		October 24th, 2020
Abdullah I. Karim **Comments:**		October 24th, 2020
Griffin L. Fluet **Comments:**		October 24th, 2020
Colleen DeMaris **Comments:**		October 24th, 2020
James West **Comments:**		October 24th, 2020

Appendix C – Document Contributions

It's worth mentioning that while each section varied in size, each team member did preliminary research that went into the decision-making process for much of this document which is a major factor considered when creating Appendix C.

Each member contributed to drafting this document evenly. Ryan Handlon created the Introduction (Section 1) and did much of the review for the document, earning a 25% share. Abdullah Karim contributed to the Functional Requirements (Section 2) and the review of the document with a percentage contribution of 25%. Colleen DeMaris wrote the Deliverables (Section 5) section of this document with a percentage contribution of 20%. Griffin created the Non-Functional Requirements (Section 3) section and had a 15% contribution to the document. James wrote out Section 4, Open Issues, Appendix A, and Appendix B which led to a 15% contribution. Blackbear Analytics as a team contributed to Appendix C to ensure transparency and talk through team dynamics.

Document B – System Design Document

Product: S.S. Media

Client: Doctor Judith E. Rosenbaum System Design Document

Black Bear Analytics

Abdullah Karim | Colleen DeMaris | Griffin Fluet | James West | Ryan Handlon

November 10, 2020

Revision History

Version Number	Release Date	Description
Version 1.0	11/10/2020	Original Release

Table of Contents

1. Introduction

The purpose of this section is to introduce the reader to both this document and the system architecture for the project, detailing introductory information that will be useful for the reader to know.

1.1 Purpose of This Document

The purpose of the Software Design Document (SDD) is to clearly outline the system design agreed upon by both the client (Doctor Rosenbaum) and Blackbear Analytics for the S.S. Media scraping tool. The Software Design will meet all requirements specified in the Software Requirements Specification (SRS). This document will cover all system design details. This includes an Architectural Design Diagram (ADD), a Technology Architecture Diagram (TAD), a Decomposition Description, Database Descriptions, File Descriptions, and a Requirements Matrix.

The intended readership of this document includes Developers, Testers, the Client, and End Users. Developers will review this document to more fully understand the system architecture and to help guide their efforts in the system's creation. Testers can use this document to help guide them in their testing of the application helping guarantee proper functionality of the product. The Client will read this document to have an understanding of the design of the product commissioned as well as guarantee that their development team understands what is needed out of the product. The end user would review this document to better understand the product's system design.

1.2 References

Banga, S. (2020). *What is Web Application Architecture? Components, Models, and Types*. Hackr.io.
https://hackr.io/blog/web-application-architecture-definition-models-types-and-more.

Fulton, J. (2018, December 12). *Web Architecture 101*.
https://engineering.videoblocks.com/web-architecture-101-a3224e126947.

Kerins, I. (2019, July 4). Solution Architecture Part 5: Designing A Well-Optimised Web Scraping Solution.
https://blog.scrapinghub.com/solution-architecture-part-5-designing-a-solution-estimating-resource-requirements.

Wikimedia Foundation. (2020, October 29). *Web crawler*. Wikipedia.
https://en.wikipedia.org/wiki/Web_crawler.

Other References:
Black Bear Analytics SRS Document

UI Design :
https://www.figma.com/file/qc0MoZOfnAJrwfYVoisxcN/BlackBearAnalytics_Mockup?node-id=0%3A1

AWS : https://aws.amazon.com/

2. System Architecture

The system architecture describes the flow, processing, and storage of information within the S.S. Media product. The architecture below includes an Architecture Design Diagram for a Scraper System as well as an accompanying Technology Architecture Diagram which illustrates the technology being used at each step in the design process. A Design Class Diagram is included in Section 2.2 which illustrates the object-oriented functionality of S.S. Media with descriptions of each step in the process.

2.1 Architectural Design

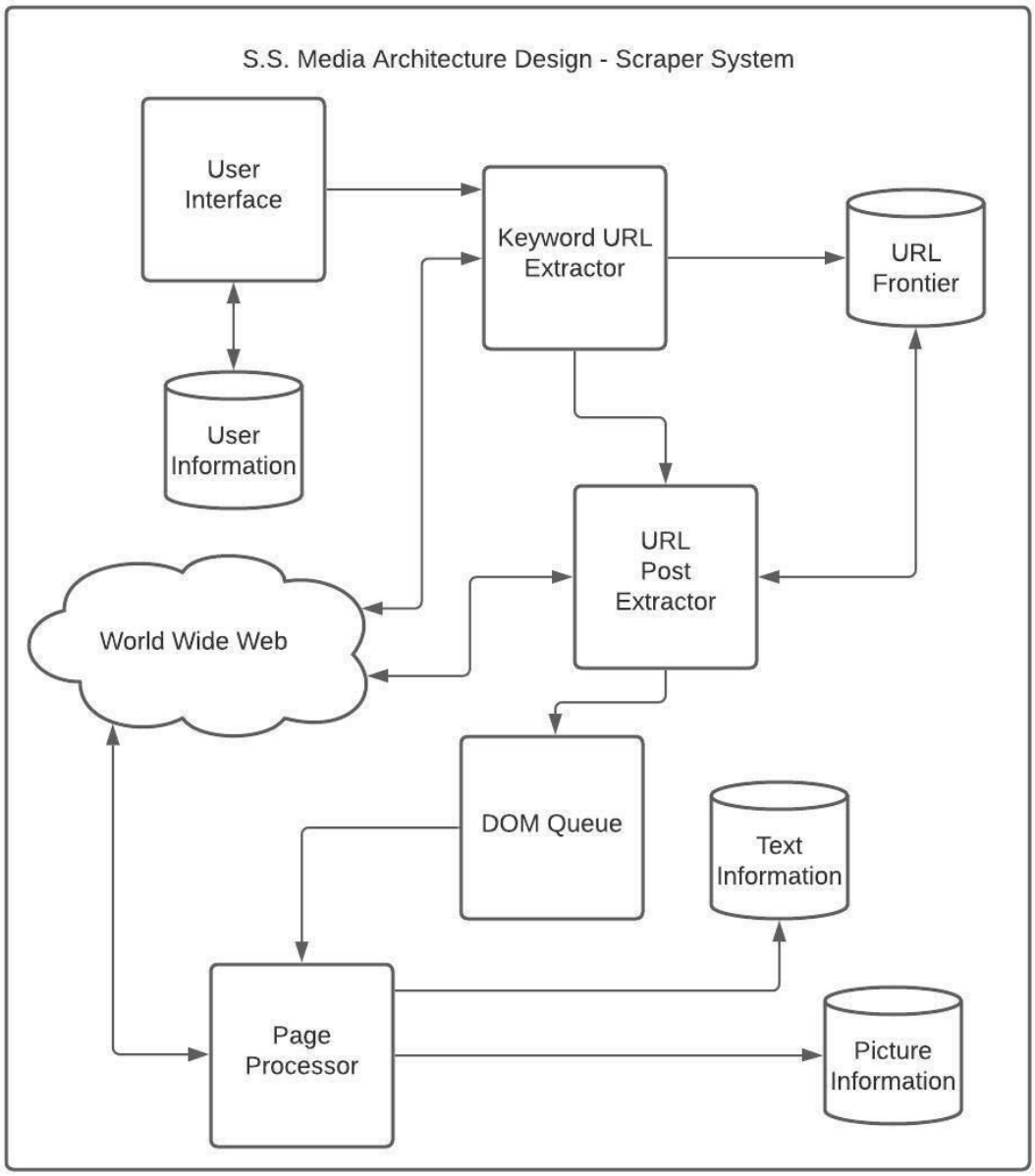


Figure 2.1: S.S. Media Architecture Design

Figure 2.1 shows the S.S. Media Architecture Design. As can be seen in the diagram, the Architecture starts with the User Interface, which manages user information and sends a requested scraped to the Keyword URL Extractor. The Keyword URL Extractor populates the URL Frontier, with URLs needed to be scrapped. From there the URL Post Extractor takes posts from the URL Frontier and creates Document Object Model (DOM) Objects which are added to the DOM Queue. The DOM Queue one by one sends DOMs to the Page Processor. There, the DOM's get parsed for relevant information and which is added to the text and picture information databases.

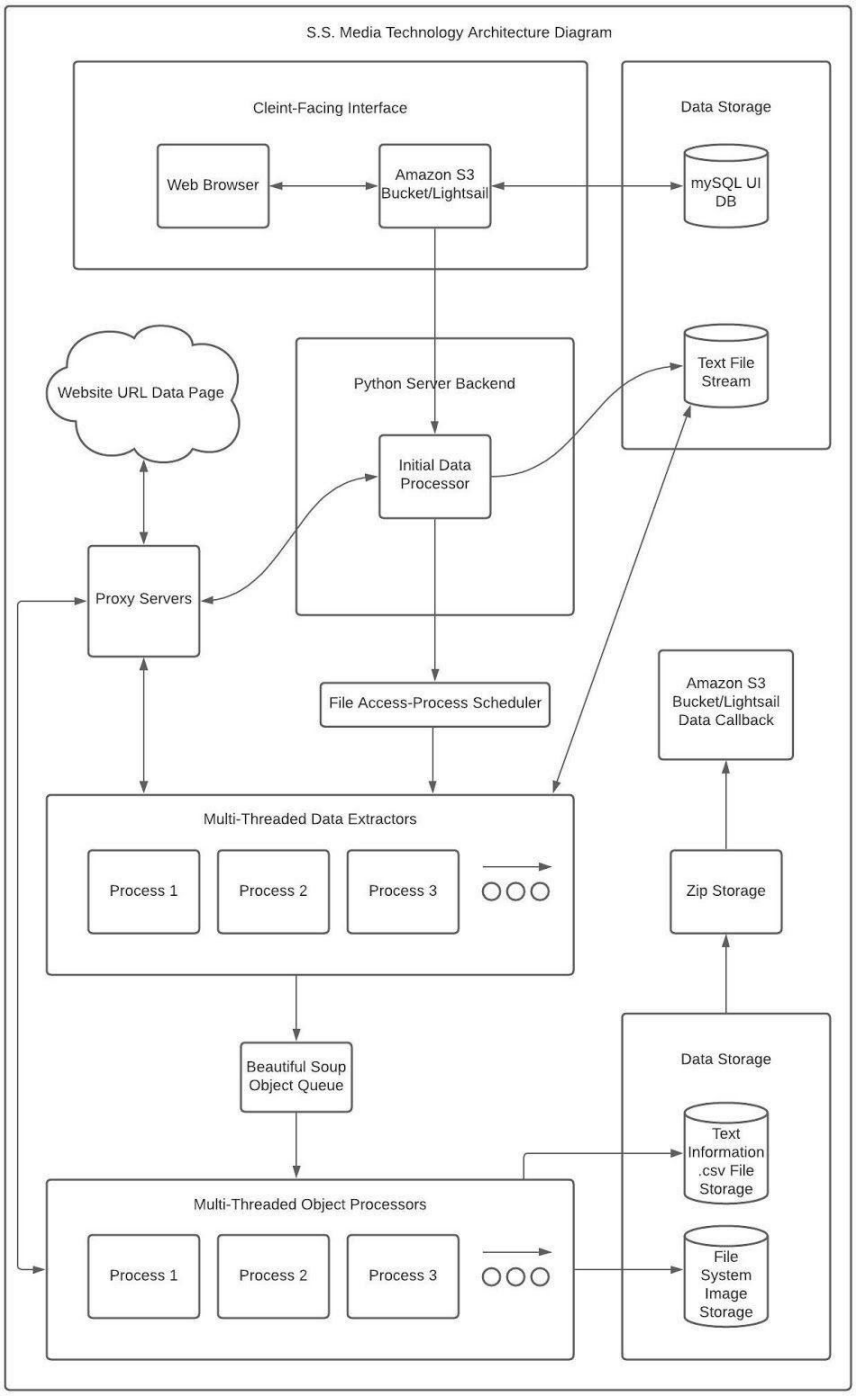


Figure 2.2: S.S. Media Technology Architecture Diagram

Figure 2.2 depicts the specific technologies that play into each part of S.S. Media's design. The user interface a client sees is accessed through a browser hosted through Amazon Web Services (AWS). The AWS will be a Python Web App service (through the Lightsail service) that the system will use to create a page where information can be input and dynamically output. AWS also provides an S3 bucket service that allows for object storage which can be used for data storage at runtime.

Any user inputs flow through the system into a Python backend where Initial data processing occurs and the URL frontier is made into a text file which the application uses to get each post's information. Information then flows through Data Extractors which create the Beautiful Soup Objects (Document Object Models, or DOMs) and are Queued up for multiple processors to then grab and process information from. DOMs are a representation of a post that processors can use to extract post information. Note that as Processors gather information, the product connects to web pages via a proxy to maintain the integrity of the root IP should an IP ban be issued by a social media platform.

Information is then stored into .csv files if they are text and into an image file system otherwise. The storage units are zipped into a file that the client can access via the user interface.

2.2 Decomposition Description

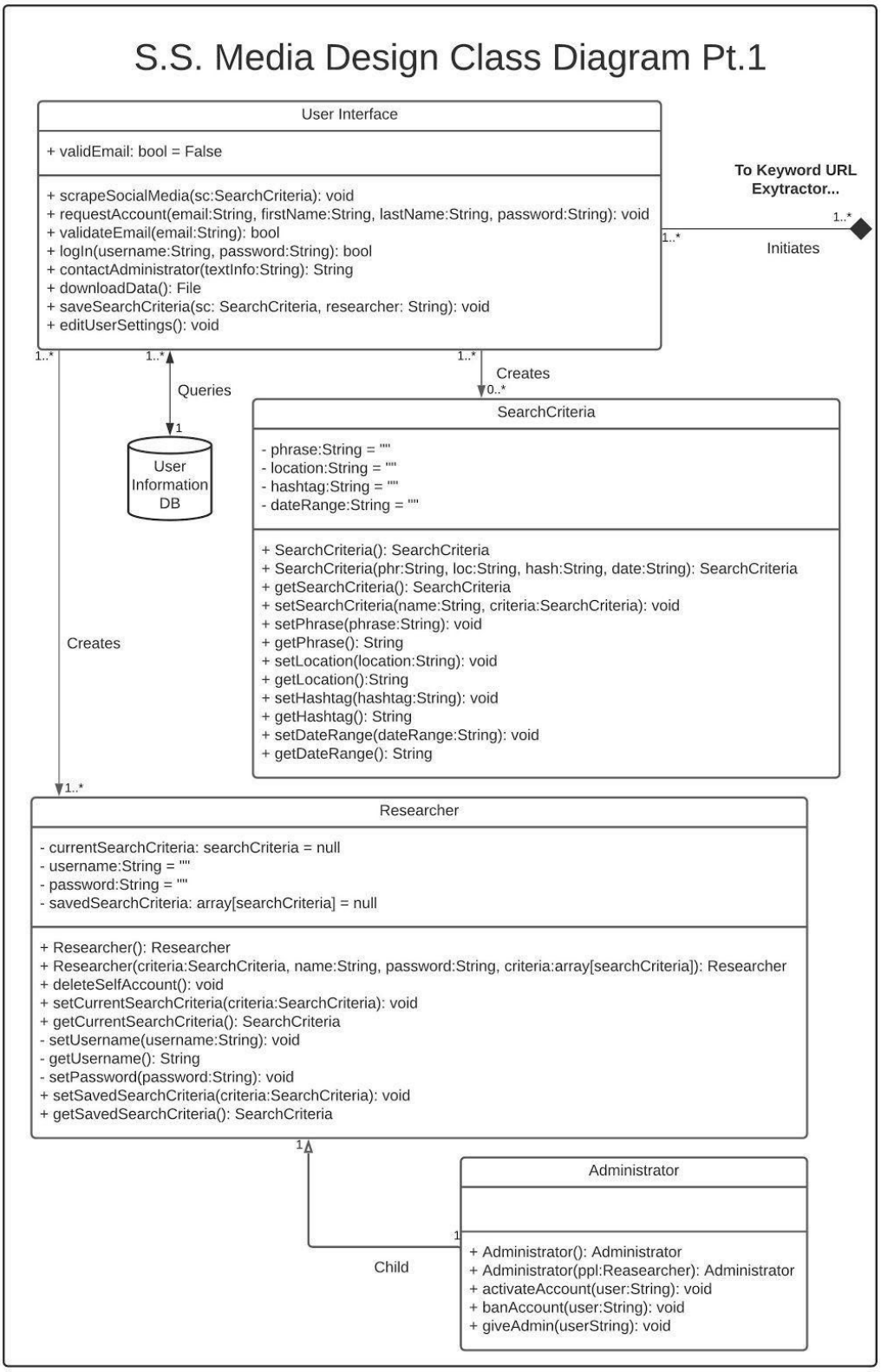


Figure 2.3: S.S. Media Design Class Diagram Part 1

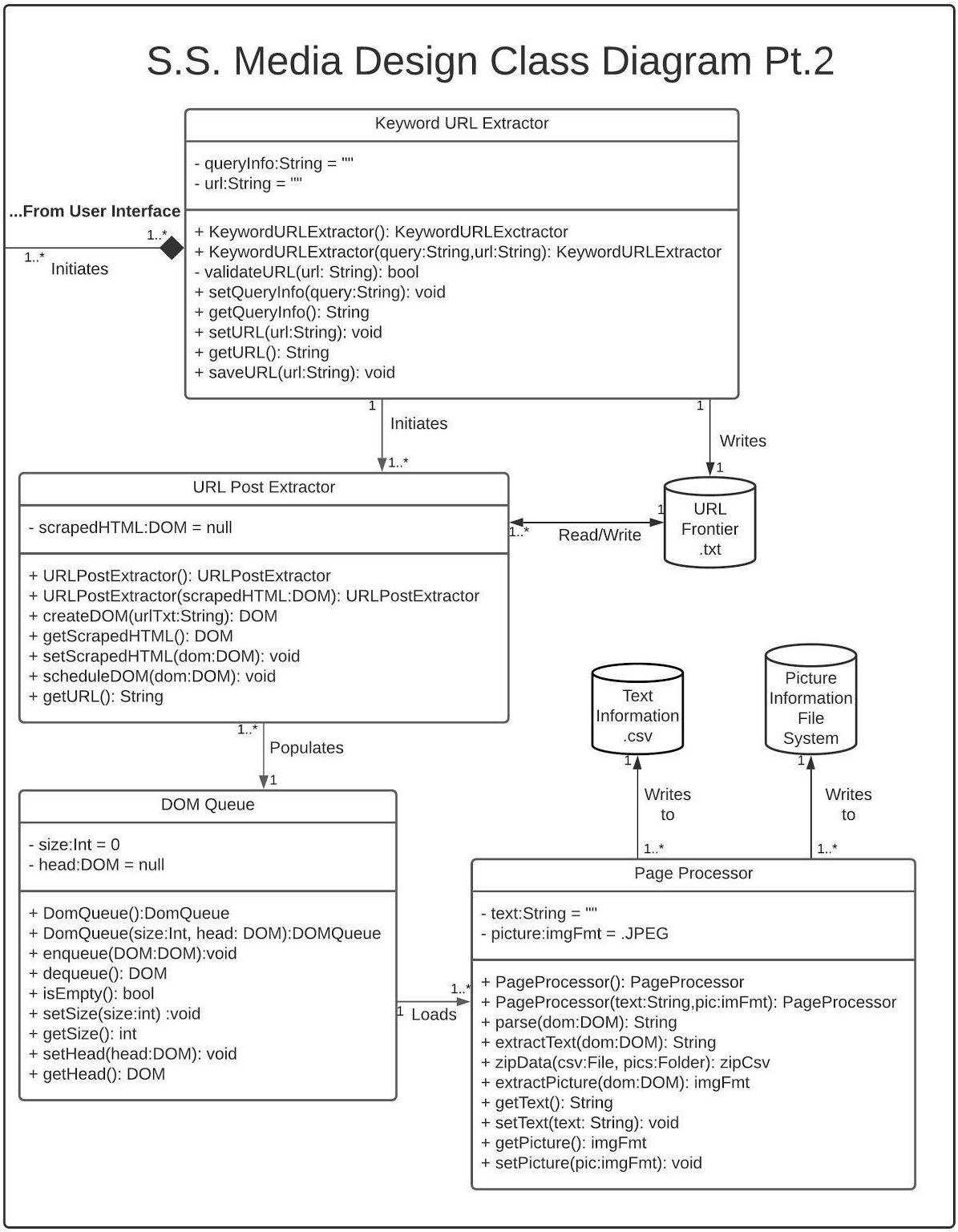


Figure 2.4: S.S. Media Design Class Diagram Part 2

Figures 2.3 and 2.4 show the planned S.S. Media Design Class Diagram (DCD). It has 11 main components which are: the User Interface, the Keyword URL Extractor, the URL Frontier file, the

URL Post Extractor, the DOM Queue, the Page Processor, the Text Information .csv file , the Picture Information File System, the User Information Database, the Search Criteria Object, and the Researcher Object. The following paragraphs each describe a main component into more detail including information such as what their job is and how they will operate. The storage will not be included in this description as they are described above and/or in Section 3.

The User Interface will be the only thing the user will ever interact with. It's main purpose is to manage the user and their information as well as initiate the scraping process. It will allow the user to do tasks such as log in, contact an administrator, select search criteria, save search criteria, start a scrape, etc. The User interface will be able to store information such as usernames, passwords, saved scrape criteria, settings, etc. in a User Information Database. Upon the start of a scrape, the User Interface will send the user's search criteria to the Keyword URL Extractor for processing.

The Keyword URL Extractor will be used to handle the front lines of the S.S. Media's scraping. After receiving a set of search criteria from the User Interface, it's job is to create a URL Frontier, a text file of the URLs of each post to be scraped later. This will be done by creating a URL associated with the specified search criteria and scraping that URL's HTML. The Keyword URL Extractor then parses the HTML for links to posts that need to be scraped and validates those links. After validation, each link gets added to the URL Frontier Database.

The URL Post Extractor's job is to extract post HTML and send it to the DOM Queue (Document Object Model). The URL Post Extractor will iterate through the URL Frontier. For each URL, it will visit the social media post's web page, scrape it's HTML, build that HTML into a DOM Object, send the DOM Object to the DOM Queue, and remove the URL from the URL Frontier.

The DOM Queue's job is to manage DOM Objects while the URL Post Extractor and Page Processor run. As they run, it will receive DOM Object's from the URL Post Extractor and add them to the end of the Queue while concurrently sending DOM Objects at the head of the Queue to the Page Processor.

The Page Processor's job is to parse the DOM Objects. It will go through all information in the DOM Object and scrape the information that is requested. If this includes links to pictures, the Page Processor will download the pictures from the internet. The scraped information will be saved into a Text Information .csv file. The pictures will be saved to a Picture information folder with a unique ID number as the file name linking it to its corresponding text information.

The Search Criteria Object manages a user's specified search criteria. It will hold data values such as Hashtags, Phrases, and Locations. It will also include functions to manipulate those parameters, create Search Criteria Objects, and save search criteria for later use.

The Researcher Object will be used to manage user accounts. It will hold data values such as current search criteria, username, password, and saved search criteria. It will also include functions to manipulate those parameters, create a new Researcher Object, and delete the Researcher Object. The Researcher Object also has a child Administrator Object. The Administrator Object contains all functionality of the Researcher Object, but also includes functions to create an administrator account, approve user accounts, edit account names, ban accounts, and give administrator privileges to user accounts.

3. Persistent Data Design

The Persistent Data Design section details how system databases and files used by S.S. Media will be designed. Each database and file description will also include a diagram for further detail.

3.1 Database Descriptions

The database structure consists of one SQL database, containing {x} tables. The first table is the user table, with its purpose being to contain any information necessary in order to validate/invalidate a user when they try to log in. This table has columns for a name, username, password (hashed), type of account (admin, basic), and email address. The string variables will be stored in a VARCHAR format, which dynamically adjusts to the length of the string. The size of a VARCHAR is the length of the data stored, plus two bytes.

Table 3.1 - User Table Schema

Name of Field	Data Type	Max Data Size	Description
Primary Key: unique_ID	String: VARCHAR	22 bytes	The primary key of the table. Each user has a specific key randomly generated upon account creation.
first_name	String: VARCHAR	22 bytes	The first name of the user.
last_name	String: VARCHAR	22 bytes	The last name of the user.
hashed_password	String: VARCHAR	52 bytes	The password of the user, chosen by them and encrypted before being stored.
account_type	int	4 bytes	The type of account for the user. Basic (represented by 0) accounts can adjust personalization settings and can run search criteria and gain results. Admin (represented by 1) accounts can do everything that Basic accounts can, as well as assign other users as Admins.
Foreign Key: email_address	String: VARCHAR	52 bytes	The email address of the user, in order to validate the identity of the person if they forget their password or type it wrong too many times. This is also used for the user to log into their account, rather than a username.

The next table is dedicated to saved search criterias. The purpose of this table is to keep a record of all the searches that want to be saved. If the user wants to make a search happen once a day every day for a week, this is where the information gets stored in order to tell the program when to search. This table includes hashtags, locations, platform being scraped, and date range.

Table 3.2 - User Input Schema

Name of Field	Data Type	Max Data Size	Description
Primary Key: email_address	String: VARCHAR	52 bytes	The primary key of the table. Each search criteria has a specific key based upon a user's email address
hashtag(s)	String: VARCHAR	252 bytes	This is the list of hashtags in the saved search. They are separated by the hashtag symbol and are in a String format. For example, #nike#boycott#boycottnike
location(s)	String: VARCHAR	252 bytes	The locations of the places in the media being scraped
platform	String: VARCHAR	22 bytes	The specific platform to scrape, either Twitter or Instagram
date_start	timestamp	8 bytes	The start date of the search, as a timestamp
date_end	timestamp	8 bytes	The end date of the search. If no end date is specified, then the search is a one-time search
scrape_frequency	int	4 bytes	The frequency of the scrape between the start and end times.
scrape_interval	time	8 bytes	The amount of time between each scrape.

3.2 File Descriptions

The files used by the system include a JSON file, text file, a picture file, and a csv file. The JSON file is used to store the user configuration settings. These settings will be set up as labels with an associated text value. The settings text will be read into the system as Strings that vary in length or as a Boolean value, depending on the setting. The following include the configurable settings with the label and default text associated with the label.

Table 3.3 - Settings

Label	Data Type	Default Value	Size
download_location	String	C:\Downloads	256 bytes
view_scrape_history	Boolean	true	1-bit
advanced_search_default	Boolean	true	1-bit
email_notifications	Boolean	true	1-bit
email	String	User's initial sign-up email	256 bytes

The text file being used by the system is a file that will temporarily store URL links in a URL Frontier. This file will be used as an intermediary to pass data from our URL Extractor to our URL Post Extractor.

Table 3.4 - Text Data Format

URL	Data Type	Size
twitter.com/example1	String	512 bytes
twitter.com/example2	String	512 bytes

The Picture file will be used to collect and organize all pictures stored in a given scrape. The Picture file will be pushed as a download to the user's configured download location upon completion of a scrape. Each picture will be saved with a unique name corresponding to it's scraped text information in the .csv file.

Table 3.5 - Picture Data Format

Picture File	Data Type	Size
pigtureID#1919193.png	Varying (.png,.jpg,.jpeg,.gif,etc)	Up to 12mb
pigtureID#1903577.png	Varying (.png,.jpg,.jpeg,.gif,etc)	Up to 12mb

The .csv file will be used as a final way to condense, sort, and organize the data collected. This .csv file will be pushed as a download to the user's configured download location upon

completion of a scrape. The scraped data .csv file structure consists of n columns depending on the website.

Table 3.6 - Text File Structure

Twitter Headers	Data Type	Size	Desc.	Instagram Headers	Data Type	Size	Desc.
ID	String	128 bytes	The tweet's ID	ID	String	128 bytes	The instagram post's ID
Date	int	4	Date tweet was posted	Date	int	4	Date photo was posted
Time	int	4	Time tweet was posted	Time	int	4	Time photo was posted
Timezone	String	128 bytes	Timezone tweet was posted in	Timezone	String	128 bytes	Timezone photo was posted in
User_ID	String	256 bytes	User's 'handle' that posted the tweet	User_ID	String	256 bytes	User's 'handle' that posted the photo
Username	String	256 bytes	User's display name	Likes	int	16 bytes	Number of likes
Tweet	String	560 bytes	Tweet contents	Replies	String	512 bytes	Content of replies
Replies	String	560 bytes	Replies' content	Location	String	256 bytes	Location the user was at when posting the photo
Retweets	int	16 bytes	Number of times retweeted	Hashtags	String	256 bytes	Hashtags used in the description
Likes	int	16 bytes	Number of times liked	Link	String	512 bytes	Link to the post
Location	String	256 bytes	Location	Image_ID	UUID	128 bytes	An ID for

			the user was at when the tweet was posted				the photo associated with the post
Hashtags	String	560 bytes	Hashtags used in the tweet	Description	String	512 bytes	Contents of description
Link	String	512 bytes	A link to the tweet				

4. Requirements Matrix

This section links the functionality described in the SRS document with the System Components and functionality designed in this SDD document. Below is a table that illustrates requirement-component connections. Components are taken directly from the Design Class Diagram (Figure 2.3 and 2.4), so looking at that for reference will help.

Table 4.1 - Requirement-Component Connection

Functional Requirements	**System Components -> Function**
Use Case #1: Log In	User Interface -> logIn(), Researcher -> getUsername(), Researcher -> setUsername(), Researcher -> getPassword(), Researcher -> setPassword()
Use Case #2: Request Account	User Interface -> contactAdministrator(), User Interface -> validateEmail(), Administrator -> approveAccount(), Administrator -> giveAdmin()
Use Case #3: Select Search Criteria	User Interface -> ScrapeSocialMedia(), Search Criteria -> All accessor/constructors/mutators()
Use Case #4: Save Search Criteria	User Interface -> saveSearchCriteria()
Use Case #5: Contact Administrator	User Interface -> contactAdministrator()
Use Case #6: Scrape Social Media	User Interface -> scrapeSocialMedia(), Keyword URL Extractor -> All functions(), URL Post Extractor -> All functions(), DOM Queue -> All functions(), Page Processor -> All functions()
Use Case # 7: Download Scraped Data	User Interface -> downloadData(), Page Processor -> downloadText(), Page Processor -> downloadPictures()
Use Case #8: Ban Account	Administrator -> banAccount()
Use Case #9: Delete Account	Researcher -> deleteSelfAccount()
Use Case #10: Activate User Account	Administrator -> activateAccount()

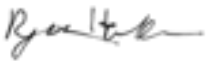
Appendix A - Agreement Between Customer and Contractor

The client (Doctor Rosenbaum) and Blackbear Analytics agree that the system's architecture meets all requirements signed in the SRS document and will be implemented in good faith. All designs will be implemented as close to the specifications in the system architecture as possible, with margins for error being slim or none. In the event that implementation must diverge from the design, the development team will notify the client as soon as possible and work together with the client until a suitable agreement is reached. If the client would like to add designs/requirements for the system's architecture after signing this document, Blackbear Analytics bears no responsibility for the failure to implement new requirements, but will attempt to add them in good faith. Should any new requirements be added, the deliverables schedule in section 2 and 3 of the SRS document will be updated immediately to reflect changes, and the designs/requirements discussed will be reflected in this document by the next deliverable deadline and changes will be finalized with the client.

Name	**Signature**	**Date Customer:**
Judith E. Rosenbaum		11/09/20

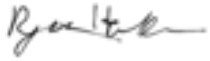
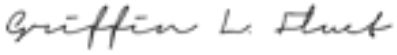
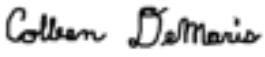
Comments:

Team:

Ryan M. Handlon		November 8th, 2020
Abdullah I. Karim		November 8th, 2020
Griffin L. Fluet		November 8th, 2020
Colleen DeMaris		November 8th, 2020
James West		November 8th, 2020

Appendix B - Team Review Sign-off

By signing your name below, you acknowledge that you are a member of Blackbear Analytics and have read the document with an in-depth comprehension of the scribed materials. You agree to complete all designs stated on this document as is, in good faith. Should any new designs come up, you agree to assist in re-drafting this document for approval as described in Appendix A. You agree that it is not required to complete designs added after this version is released, but understand that all designs, new or old, must be implemented in good faith.

Ryan M. Handlon		November 8th, 2020
Abdullah I. Karim		November 8th, 2020
Griffin L. Fluet		November 8th, 2020
Colleen DeMaris		November 8th, 2020
James West		November 8th, 2020

Comments:

Appendix C - Document Contributions

Each member contributed to drafting this document evenly (20% each). Ryan Handlon created the Introduction (Section 1), aided in the creation of the System Architecture (Section 2.1), made the description for the Decomposition Description (Section 2.2), and helped in the revision process. Abdullah Karim helped in the creation of the Architectural Design Diagrams (Section 2.1), the development of the Decomposition Diagram (2.2), and the Requirements Matrix (Section 4). Griffin Fluet created the Design Class Diagram (Section 2.2), and helped in the revision process. Colleen DeMaris created the Persistent Data Design (Section 4) and helped in the revision process. James West also created the Persistent Data Design (Section 4) and helped in the revision process. The Appendices were appended and modified from the SRS document and Appendix C have been read and reviewed by the whole team.

Document C – User Interface Design Document

Product: S.S. Media

Client: Doctor Judith E.Rosenbaum

User Interface Design Document

Black Bear Analytics

Abdullah Karim | Colleen DeMaris | Griffin Fluet | James West | Ryan Handlon

November 24, 2020

Revision History

Version Number	Release Date	Description
Version 1.0	11/24/2020	Original Release

Table of Contents

1. Introduction

The purpose of this section is to introduce the reader to both this document and the system architecture for the project, detailing introductory information that will be useful for the reader to know.

1.1 Purpose of This Document

The purpose of the User Interface Design Document (UIDD) is to clearly outline the user interface design agreed upon by both the client (Doctor Rosenbaum) and Blackbear Analytics for the S.S. Media scraping tool. The User Interface will meet all requirements specified in the Software Requirements Specification (SRS). This document will cover all user interface design details. This includes User Interface Standards, a Navigation Diagram, a User Interface Walkthrough, Data Validation, and a Report Formats.

The intended readership of this document includes Developers, Testers, the Client, and End Users. Developers will review this document to more fully understand the system user interface and to help guide their efforts in the system's creation. Testers can use this document to help guide them in their testing of the interface helping guarantee proper functionality of the product. The Client will read this document to have an understanding of the design of the product commissioned as well as guarantee that their development team understands what is needed out of the product. The end user would review this document to better understand the product's user interface design.

1.2 References

Black Bear Analytics SRS Document Black Bear Analytics SDD Document

UI Design :
https://www.figma.com/file/qc0MoZOfnAJrwfYVoisxcN/BlackBearAnalytics_Mockup?node-id=0%3A1

2. User Interface Standards

The User Interface Standards section will go over the design standards and themes that are used throughout the user interface. This section also touches on the general error handling.

As seen in Figure 2.1 and 2.2, the drop down boxes are of the same component and will display an example of input in the field when the user has not selected an option yet. Clickable buttons will only be made clickable when valid input in the corresponding text field(s) is entered, until then the button will be greyed out and not be able to be clicked. Once valid input is entered into every field needed the button will be highlighted and clickable. Toggles will be highlighted when clicked to represent that they are selected and greyed out when not. Labels for buttons and text fields will be descriptive as to what the button does when clicked or what is intended to be entered in the text field. Text fields also provide an example of possible input above the field. Navigation of the user interface will be discussed later in this document. All of the themes listed above are common components within the user interface and will be used throughout.

General error handling will typically attempt to send the user back to the last screen they were at with all of the valid inputs, if any, that the user had entered before the error occurred. If this is not possible due to corrupt or invalid data, the user will be sent back to the home page, and the user will need to manually enter valid input data again. For example, if the user were to hit an error during the Search Criteria stage, having entered the Platform and a single Hashtag they wished to look for, then the user would be sent back to the Search Criteria stage with the selected platform still selected, and the Hashtag still entered. The user will make another attempt to enter valid inputs, and continue on without further interruption. If the user's search criteria data, such as the selected platform and hashtag, were to be corrupted, the user would then be sent back to the home page, without any inputs previously entered being saved. The user would then need to go back to the Search Criteria page and manually re-enter any valid inputs they had before the error occurred.

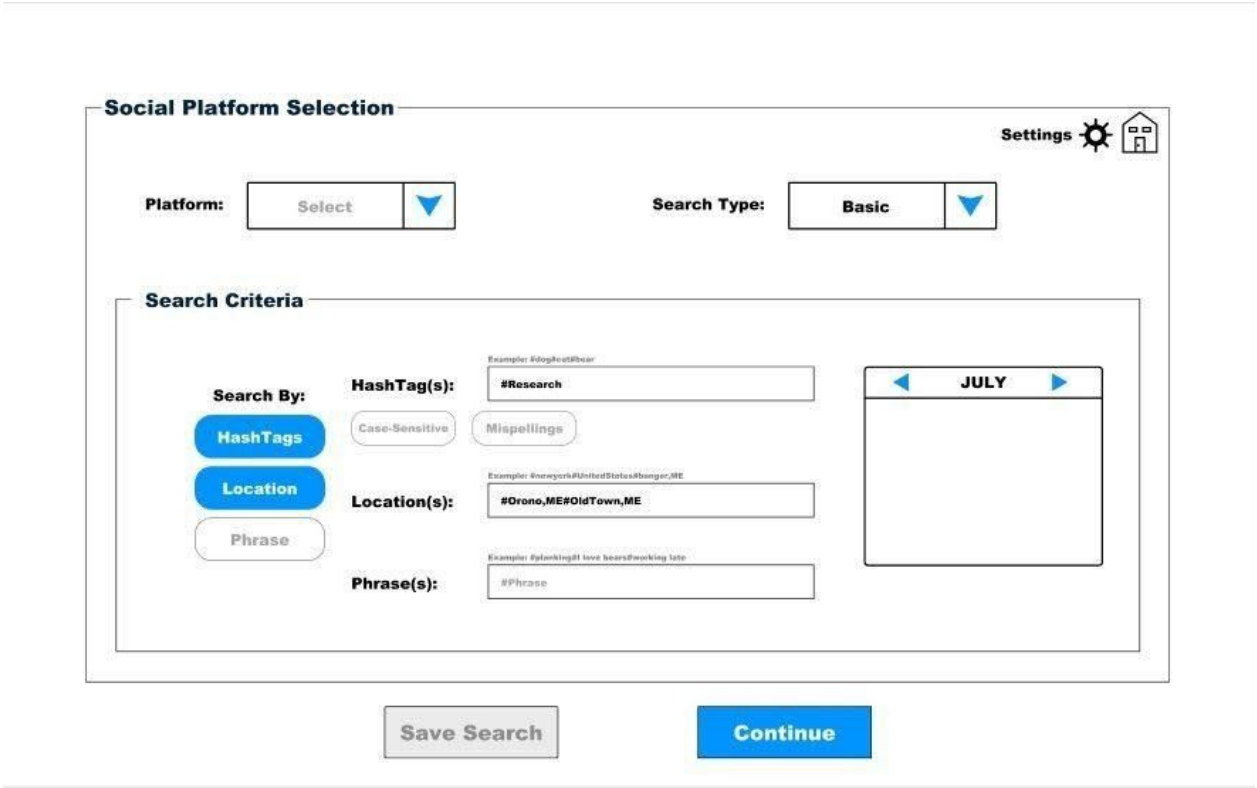


Figure 2.1: New Search Page

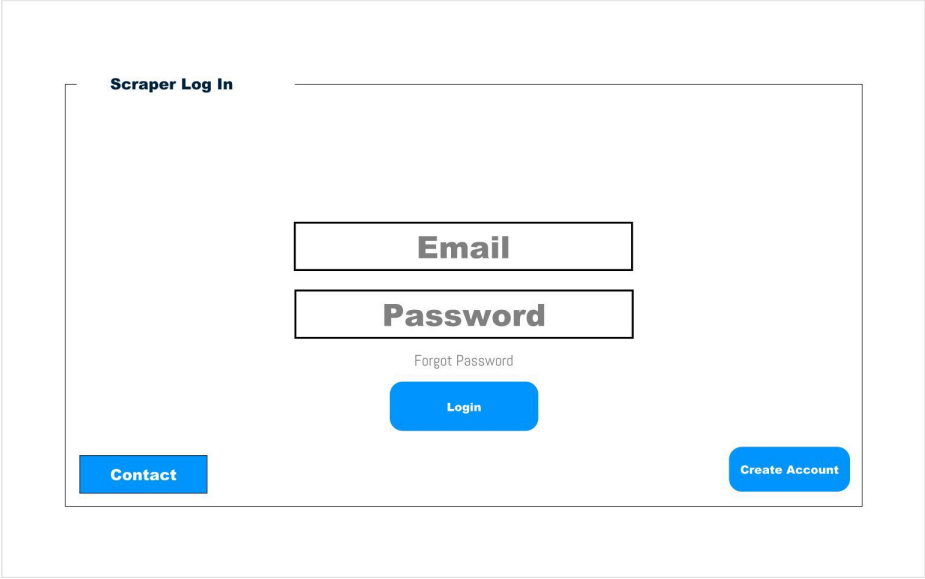


Figure 2.2: Login Page

3. User Interface Walkthrough

The User Interface Walkthrough section is intended to help visualize what the website will look like. It will include a Navigation Diagram detailing which pages can navigate to which. Following that will be a walkthrough of each unique page on the website including information on how to navigate to and from the page, what the purpose of the page is, it's main functions, and descriptions of the functionality of all buttons and fields on the page.

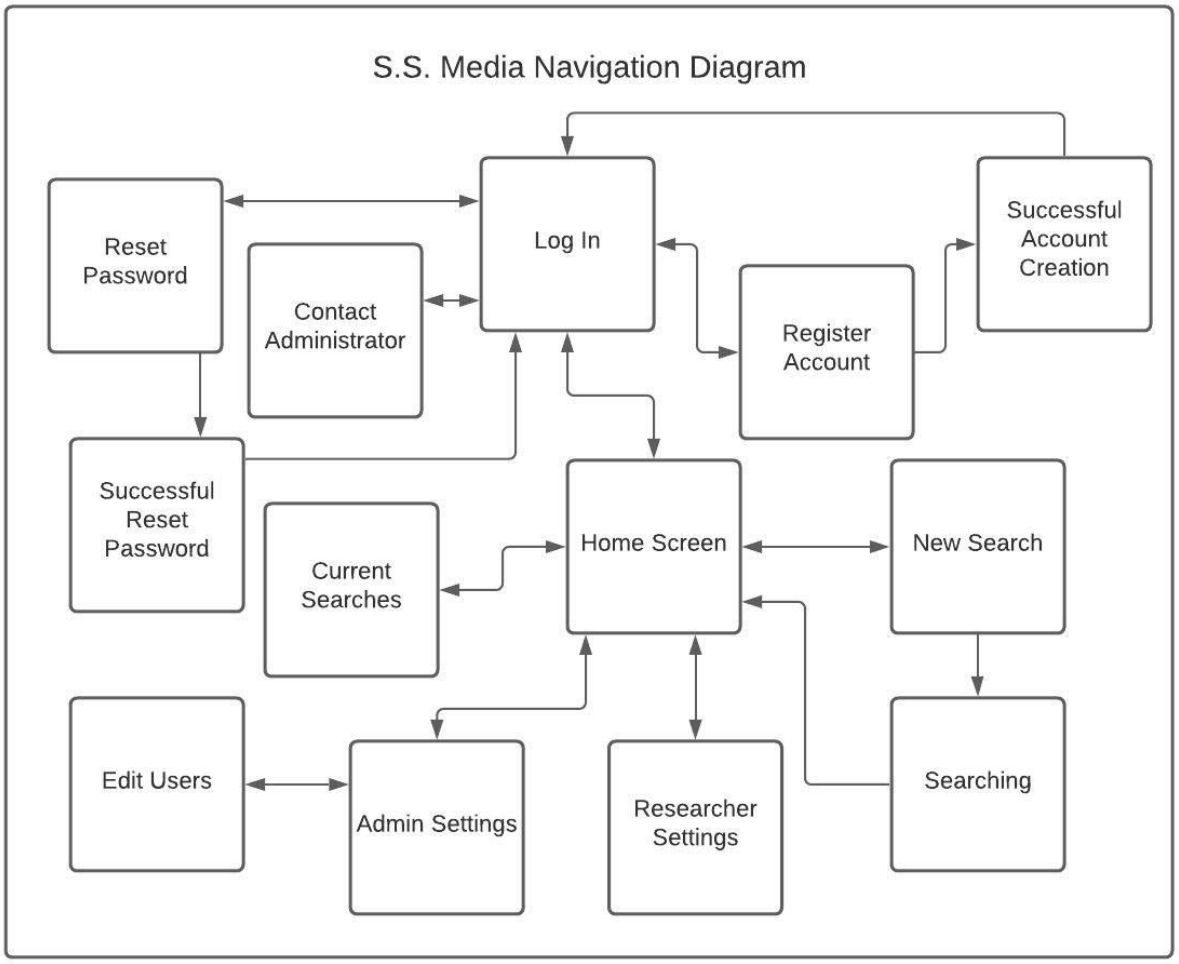


Figure 3.1: S.S. Media Navigation Diagram

The S.S. Media Navigation Diagram, as seen in Figure 3.1, displays all pages that can be visited by the user on the scraping website. When a user navigates to the website for the first time, they will be brought to the login page. From there they will be able to navigate to any other page by the way of buttons. The Navigation Diagram shows all relationships between the pages. An arrow pointing from one page to another indicates that there is a way to navigate to the pointed at page from the current page. The following page descriptions will show a mockup of each page,

including how to navigate to and from the page, what the purpose of the page is, it's main functions, and descriptions of the functionality of all buttons and fields on the page.

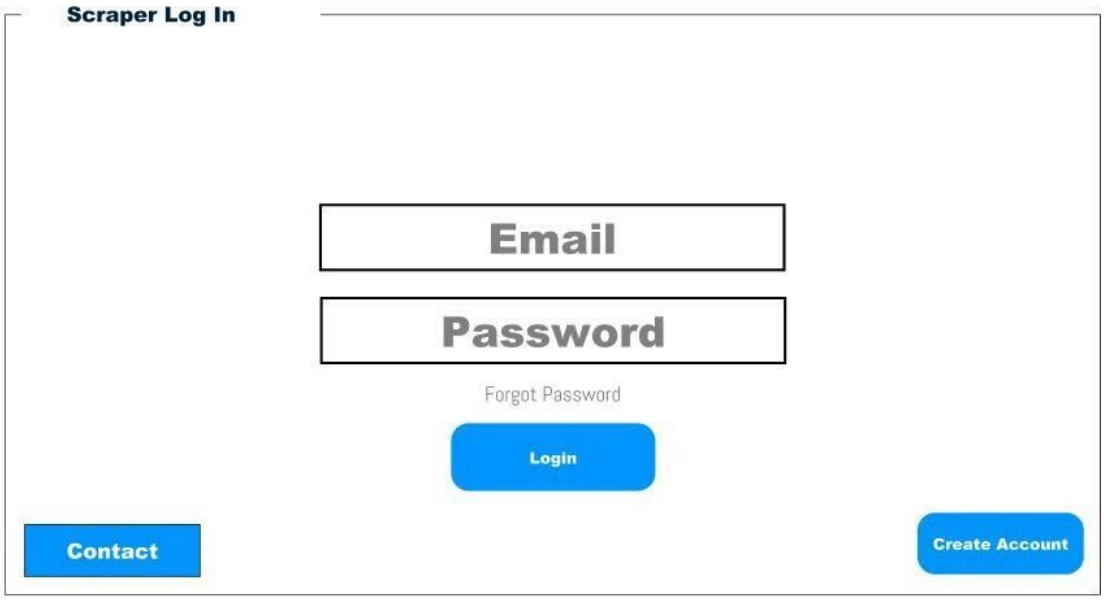


Figure 3.2: Login Page

The Login Page, Figure 3.2, is the first screen a user will see upon entering the site. It is navigated to either as the first page a user would view upon viewing the website, or via the buttons on the Reset Password, Successful Reset Password, Contact Administrator, Register Account, Successful Account Creation, or Home Screen Pages. It's main purpose is to log into a user's account, but includes extra functionality. It includes 4 buttons and 2 text fields.

- Email: This text field will take email text input from the user.
- Password: This text field will take password text input from the user.
- Forgot Password: This button will navigate the user to a new page that will include instructions on how to recover their password.
- Login: This button will take the text inputs from the Username and Password text fields and attempt to use them to log into a user's account. Should login be successful, the user will be brought to the home screen. If it is unsuccessful an invalid username/password message will be displayed.
- Create Account: This button will navigate the user to the Account Creation Page.
- Contact: This button will navigate the user to the Contact Administrator Page.

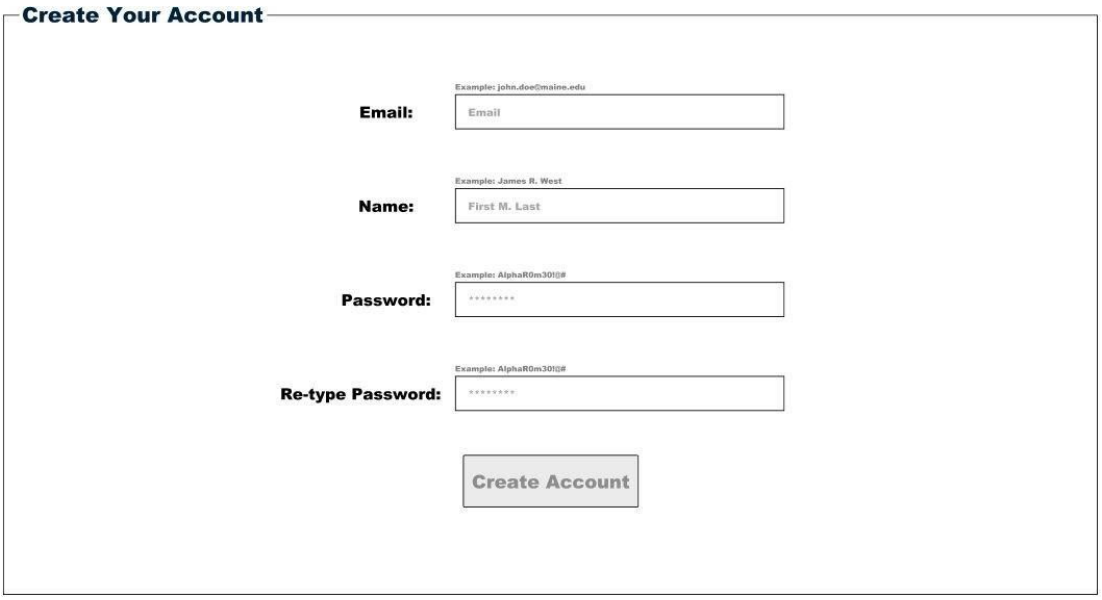


Figure 3.3: Create Account Page

The Create Account Page, Figure 3.3, is navigated to upon clicking the Create Account button on the Login Page. It's main purpose is to gather information required for a new user account and includes 4 text fields and 1 button.

- Email: This text field will take email text input from the user.
- Name: This text field will take name text input from the user.
- Password: This text field will take password text input from the user.
- Re-type Password: This text field will take retyped password text input from the user.
- Create Account: This button will take the information inputted in this page's text fields and check for validity. If information is not valid an error will be displayed to the user. If the input is valid and the two password inputs match each other, the information is saved and sent to an administrator for confirmation. The user is then sent to the Account Creation Success page.

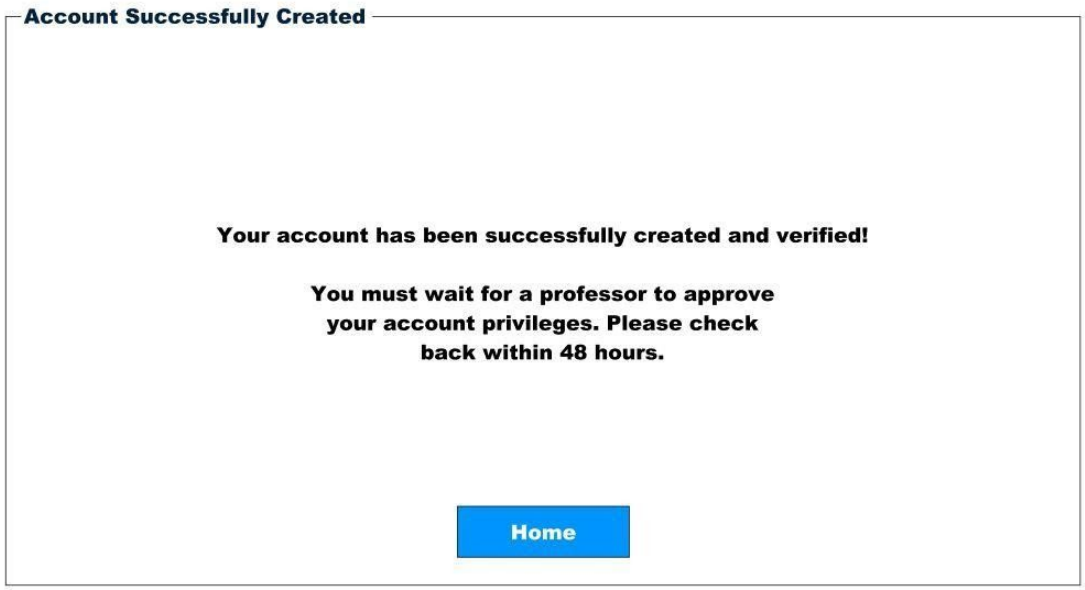


Figure 3.4: Account Creation Success Page

This Account Creation Success Page, Figure 3.4, is navigated to by clicking on the Create Account button on the Create Account page with valid inputs. This page's primary function is to provide a message to the user about successful account creation and waiting for account approval. The page includes 1 button.

- Home: This button will navigate the user to the Login page.

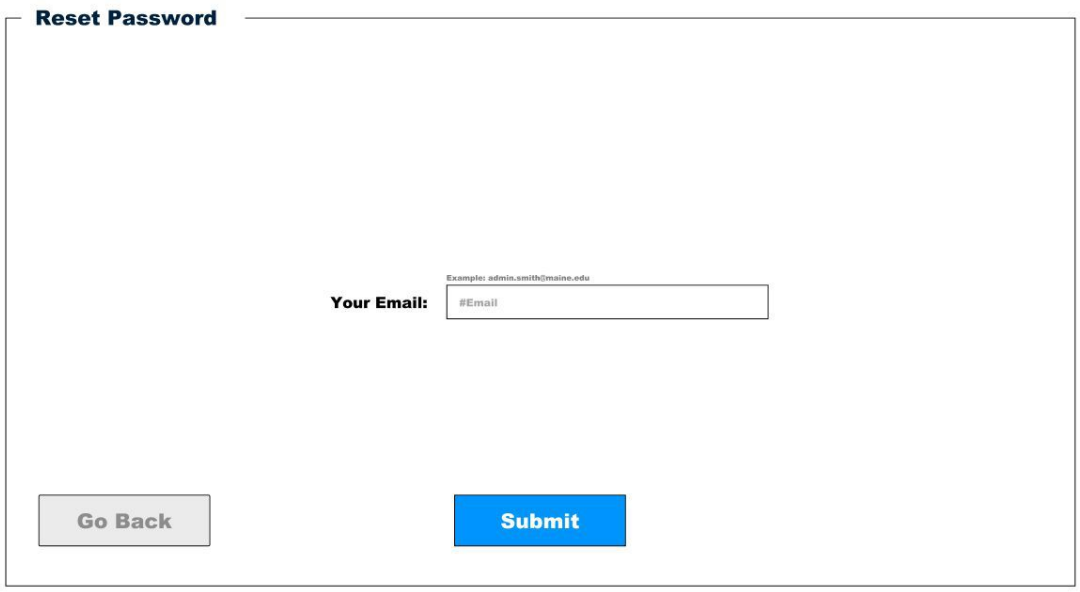


Figure 3.5: Reset Password Page

This Reset Password Page, Figure 3.5, is navigated to by clicking on the Reset Password button on the Login page. This page's primary function is to reset a user's password after they have forgotten it. The page includes 1 text field and 2 buttons.

- Your Email: This text field takes email text input from the user.
- Submit: The submit button will take a user's input from the Your Email text field and check if there is a user account with that email associated to it. If there is a match, that account's password will be reset and the user will be navigated to the reset password confirmation page.
- Go Back: This button will navigate the user back to the Login page

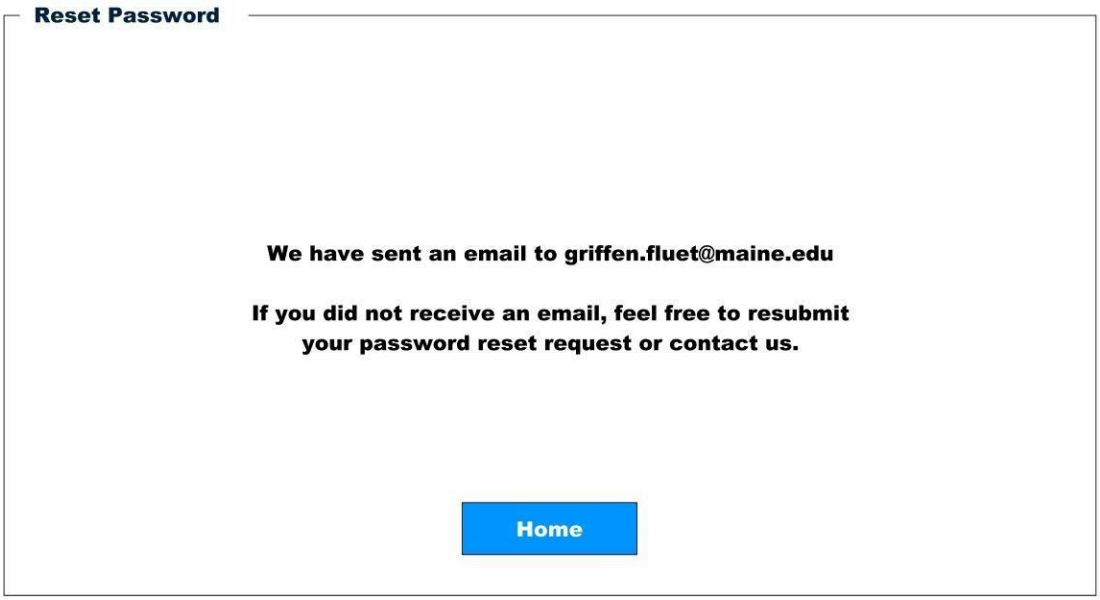


Figure 3.6: Reset Password Success Page

This Reset Password Success Page, Figure 3.6, is navigated to by clicking on the Submit button on the Reset Password page with valid input. This page's primary function is to provide a message to the user directing them to their email for further instruction. The page includes 1 button.

- Home: This button will navigate the user to the Login page.

Figure 3.7: Contact Administrator Page

This Contact Administrator Page, Figure 3.7, is navigated to by clicking the Contact button on the Login page. This page's primary function is to allow the user to send a message to an administrator. The page includes 2 text fields and 2 buttons.

- Your Email: This text field takes email text input from the user.
- Message: This text field takes message input from the user.
- Submit: This button takes the text input from the two text boxes on this page and sends the information as a message to an Administrator.
- Go Back: This button will navigate the user back to the Login page.

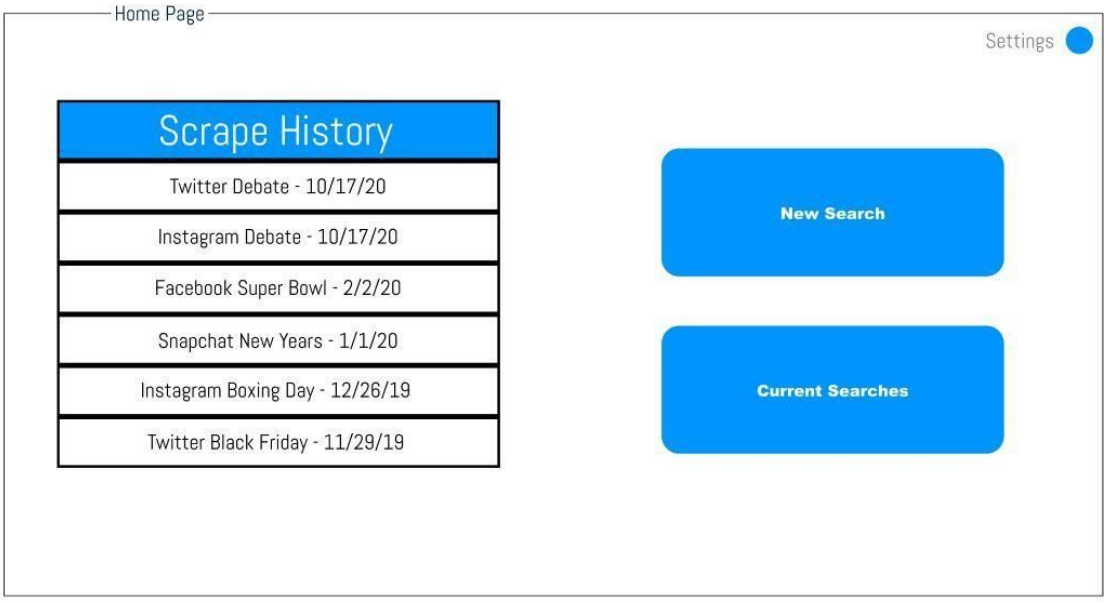


Figure 3.8:Home Page

The Home Page, Figure 3.8, is navigated to by clicking on the Login button on the Login page. It's primary function is to help navigate the user to the applications primary functions. The page has 3 buttons and one table with many buttons.

- New Search: This button will navigate the user to the New Search page.
- Current Searches: This button will only be visible if the user is logged into an Administrator account. When clicked it will navigate the user to the Current Searches page.
- Settings: This button will navigate the user to the settings page.
- Scrape History: This table will display a list of previous scrapes the user has done. Upon clicking one of these previous scrapes the user will be navigated to the new search page with that scrape's specific search results.

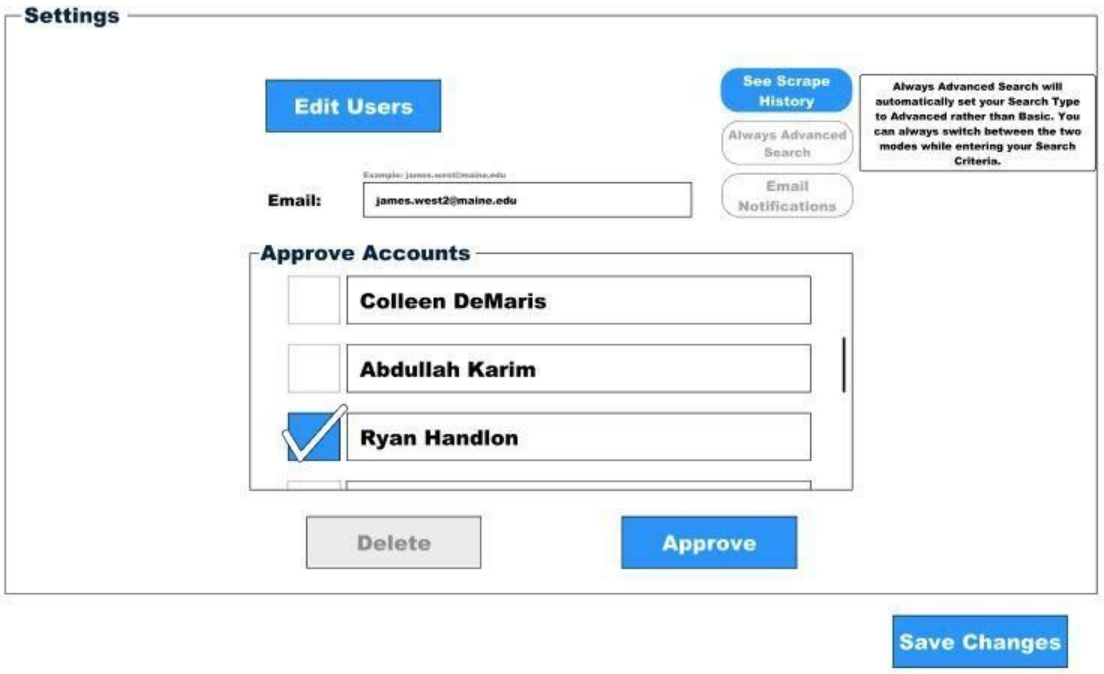


Figure 3.9: Admin Settings Page

The Admin Settings Page, Figure 3.9, is navigated to from the home page. This is the page that shows up for admin users. It allows the admin users to approve accounts, delete accounts, or to adjust personal email data and the location for the downloads. It also allows the user to set the search to an always advanced search. The user can also enable and disable email notifications.

- Edit Users: This button will bring the user to the Edit Users page.
- See Scrape History: Will enable/disable the scrape history on the home page.
- Always Advanced Search: Will enable/disable advanced search being the default search.
- Email Notifications: Will enable/disable email notifications for this user.
- Approve Accounts: This section will allow the admin user to approve or delete account requests.
- Save Changes: This button will save any changes made to the settings in a config file.

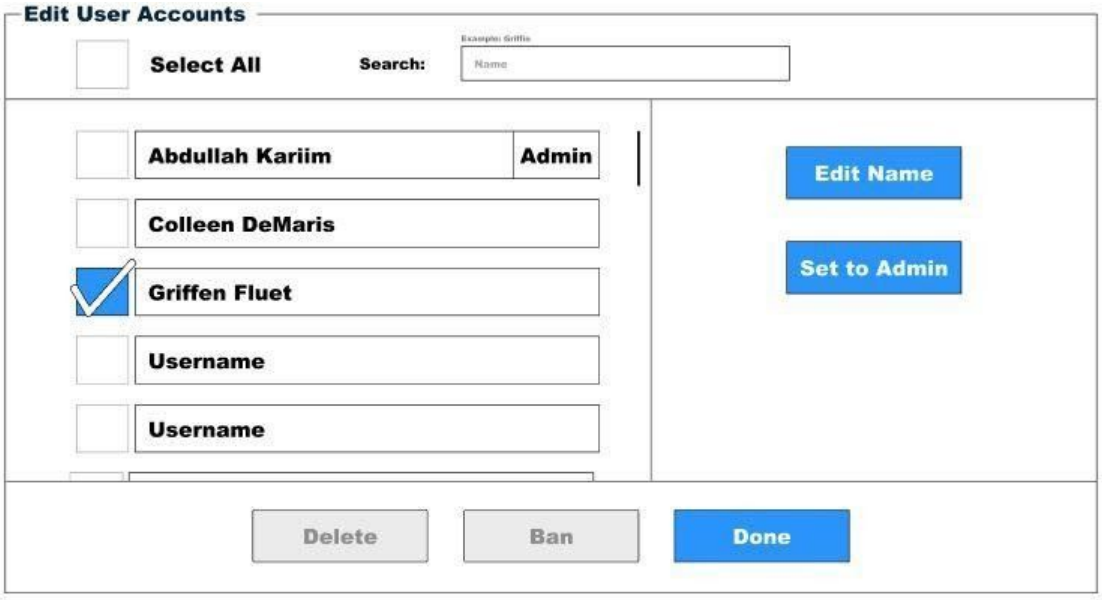


Figure 3.10: Edit Users Page

The Edit Users Page, Figure 3.10, is accessed via the Edit Accounts button on the Admin Settings page. It allows the user to delete, ban, or edit any other accounts, as well as see which accounts are admin and which are not.

- Delete: Deletes the selected account(s).
- Ban: Bans the selected account(s), but does not delete the information.
- Edit Name: Edits the name of the selected account(s).
- Set to Admin: Gives admin rights to the selected account(s).
- Done: Navigates to the previous page, the Admin Settings.

Figure 3.11: Researcher Settings Page

The Researcher Settings Page, Figure 3.11, is navigated to from the Home page. This is the page that shows up for non-admin users.

- Email: Allows the user to edit their email address (which is also used as their login).
- Download Location: Allows the user to change where the downloads go on their computer.
- See Scrape History: Will enable/disable the scrape history on the home page.
- Always Advanced Search: Will enable/disable advanced search being the default search.
- Email Notifications: Will enable/disable email notifications for this user.
- Logout: Logs out of the user's account.
- Deactivate Account: Deactivates and deletes the information of the user's account.
- Save Changes: Saves any changes made to the settings, and navigates to the previous page, the Home page.

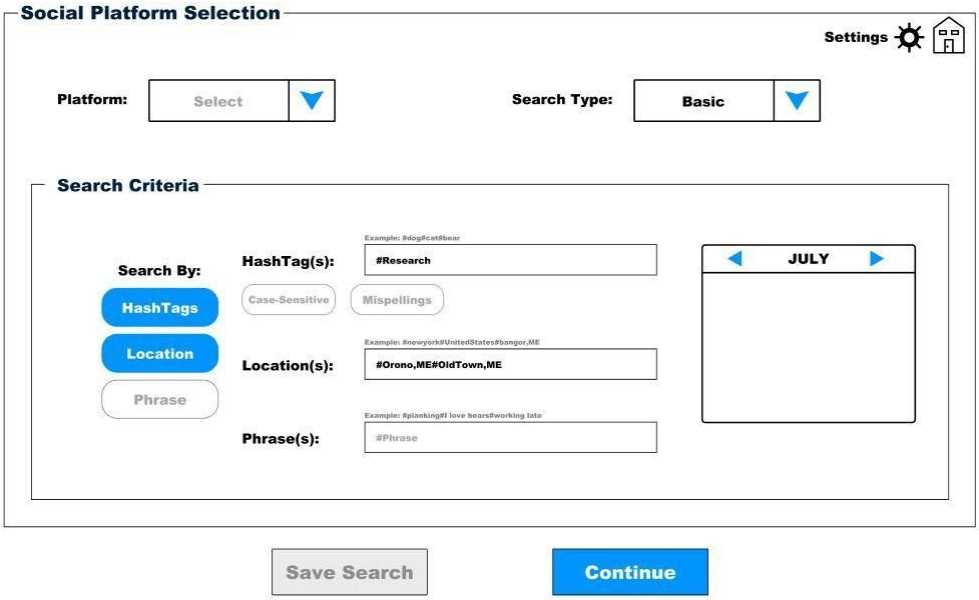


Figure 3.12: New Search Page

The New Search Page, Figure 3.12, is navigated to from the Home page. Its primary purpose is to start a new search, whether it be a basic or an advanced search. This page has two drop down menus, three different ways of searching (hashtags, location, and/or phrase), which can be searched for in any combination (as long as it includes at least one of the options), and boxes in order to fill out the search queries for the hashtags, location, or phrases.

- Platform: Allows the user to select which platform will be scraped.
- Search type: There are two options, basic or advanced. The advanced version of this will be shown on the next page.
- Hashtags button: Enables/disables search by hashtags.
- Location: Enables/disables search by location.
- Phrase: Enables/disables search by a phrase.
- Hashtags text field: Allows the user to enter hashtags to search for.
- Case Sensitive: Enables/disables case sensitivity with the search.
- Misspellings: Enables/disables letting the program look for common misspellings of the word(s).
- Location text field: Allows the user to enter locations to search for.
- Phrase text field: Allows the user to enter a phrase to search for.
- Save Search: This button saves the search to a database so that it can be searched for again or documented.
- Continue: Continues to the next page: the Searching page.
- Home: Navigates the user to the Home Page.

- Calendar: This calendar will allow the user to select a date range for the search specifications.

Figure 3.13: Searching Page

This is the Searching page, Figure 3.13. The primary purpose of this page is to show the scrape that is currently in progress, and show how many pieces of data (tweets, instagram posts, etc.) have been found.

- End Search: stops the search and finishes saving the data found.
- New Search: navigates to the Search page.
- Settings: navigates to the settings page, depending on whether the user account is an admin or a basic account.

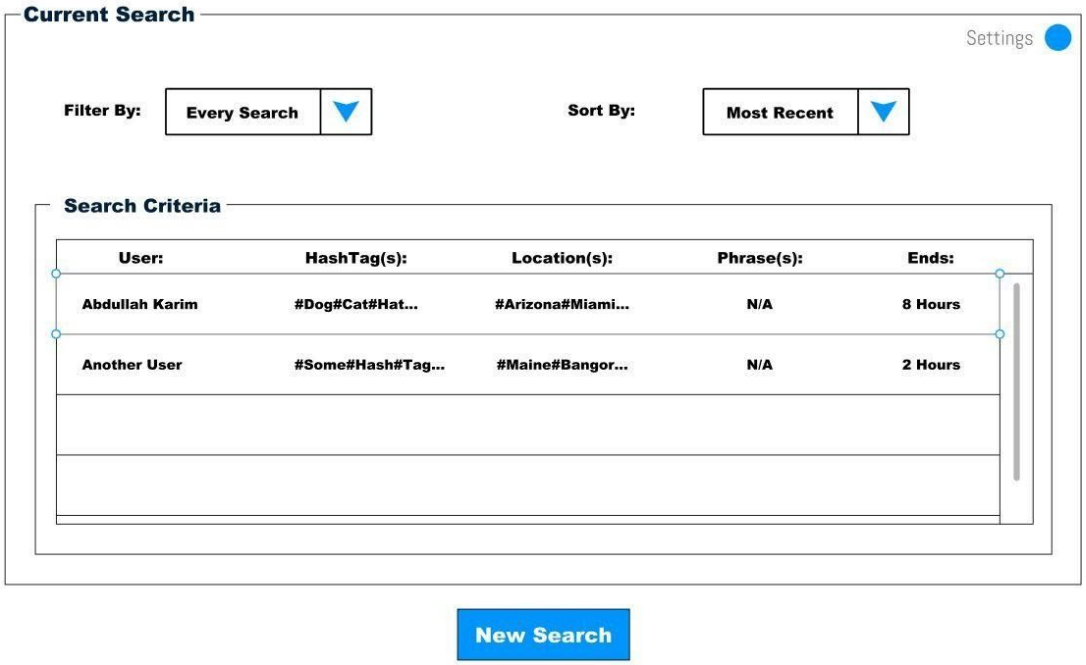


Figure 3.14: Current Search Page

The Current Search Page, Figure 3.14, is navigated to by clicking on the Current Searches button on the on the Home page. It's primary function is to display all searches being run on the platform at the current time. The Page has 2 buttons, 2 drop down boxes, and one table.

- New Search: This button will navigate the user to the New Search page.
- Settings: This button will navigate the user to the settings page.
- Filter By: This drop down box will allow the user to filter the current searches by Every Search, a Specific User, or Social Media Platform
- Sort By: This drop down box will allow the user to sort the current searches by Most Recent, Least Recent, Number of Posts Scraped, and Estimated Time to Completion.

4. Data Validation

This section shows the data a user will be able to enter into our product. It includes a unique identifier, corresponding GUI screens (see above sections), the formatting of data, and the limits for each data input.

Table 4.1: Data Input Validation Specifications

Unique ID	Data Type	Data Format	Limits	GUI Screen (s)
Email	String	[Email]@[domain].[extension]. Cannot be empty.	100 chars	1. Reset Password 2. Settings 3. Create Your Account 4. Contact Us Form 5. Scraper Log In
Download Location	String	[Drive]\[filepath]\...\[destination]. Cannot be empty.	500 chars	1. Settings
Search Users	String	Any acceptable string that follows a string regex checking if it's empty.	50 chars	1. Edit User Accounts
Contact Message	String	Any acceptable string that follows a string regex checking if it's empty.	500 chars	1. Contact Us Form
Password	String	Any acceptable string that follows a string regex checking if it's empty. Created passwords must have a capital letter, a lowercase letter, a number, and a special symbol. Passwords must also be at least 8 characters long.	40 chars	1. Scraper Log In 2. Create Your Account
Name	String	[First Name] [Middle Initial]. [Last Name]. Must follow this format.	60 chars	1. Create Your Account
HashTag(s) Field	String	Each hashtag follows the	60 chars	1. Social Platform

		format: #[string]. Multiple hashtags can be input as long as they are comma-separated.		Selection
Location(s) Field	String	Each Location can be entered as a string, with multiple locations being designated as comma-separated values.	100 chars	1. Social Platform Selection
Phrases(s) Field	String	Each Phrase can be entered as a string, with multiple phrases being designated as comma-separated values.	280 chars	1. Social Platform Selection
Run For Field	integer	Must be between 1 and the maximum signed integer limits.	10 chars	1. Social Platform Selection - Advanced Search

5. Report Formats

This section covers the hard copies that our users will get when downloading scraped information.

Our product generates a downloadable .zip file. This .zip contains a folder of all images scraped as well as a .csv file for text data. This .zip has its name made up from the keywords of the search. The name of each picture file corresponds to the unique ID of each picture in a .csv file discussed below.

A separate .csv file is located at the same level as the image folder directory. It contains all text data from a scrape. Figures of the formatting for both the .csv file and the picture file system are found below:

id	date	time	timezone	user_id	username	tweet	replies	retweets	likes	location	hashtag	Link
1	MM/DD/YYYY	23:59:59	UTC	1	#beta_tester	#test	0	0	1	Arkansas City	#test	www.test.com/postname
2	MM/DD/YYYY	23:59:53	UTC	2	#beta_tester	#test	0	0	1	Arkansas City	#test	www.test.com/postname
3	MM/DD/YYYY	23:59:49	UTC	3	#beta_tester	#test	0	0	1	Arkansas City	#test	www.test.com/postname
4	MM/DD/YYYY	23:59:45	UTC	4	#beta_tester	#test	0	0	0	Arkansas City	#test	www.test.com/postname
5	MM/DD/YYYY	23:59:44	UTC	5	#beta_tester	@test	0	0	0	Arkansas City	#test	www.test.com/postname
6	MM/DD/YYYY	23:59:42	UTC	6	#beta_tester	@test	0	0	0	Arkansas City	#test	www.test.com/postname
7	MM/DD/YYYY	23:59:41	UTC	7	#beta_tester	@test	2	0	5	Arkansas City	#test	www.test.com/postname
8	MM/DD/YYYY	23:59:37	UTC	8	#beta_tester	@test	0	0	1	Arkansas City	#test	www.test.com/postname
9	MM/DD/YYYY	23:59:37	UTC	9	#beta_tester	@test	0	0	1	Arkansas City	#test	www.test.com/postname
10	MM/DD/YYYY	23:59:36	UTC	10	#beta_tester	@test	0	0	0	Arkansas City	#test	www.test.com/postname
11	MM/DD/YYYY	23:59:34	UTC	11	#beta_tester	@test	1	1	0	Arkansas City	#test	www.test.com/postname
12	MM/DD/YYYY	23:59:31	UTC	12	#beta_tester	@test	0	0	4	Arkansas City	#test	www.test.com/postname
13	MM/DD/YYYY	23:59:30	UTC	13	#beta_tester	@test	0	2	2	Arkansas City	#test	www.test.com/postname
14	MM/DD/YYYY	23:59:27	UTC	14	#beta_tester	@test	0	0	0	Arkansas City	#test	www.test.com/postname
15	MM/DD/YYYY	23:59:25	UTC	15	#beta_tester	@test	7	12	3	Arkansas City	#test	www.test.com/postname
16	MM/DD/YYYY	23:59:22	UTC	16	#beta_tester	@test	1	0	2	Arkansas City	#test	www.test.com/postname
17	MM/DD/YYYY	23:59:22	UTC	17	#beta_tester	@test	0	0	0	Arkansas City	#test	www.test.com/postname
18	MM/DD/YYYY	23:59:21	UTC	18	#beta_tester	#test	0	0	3	Arkansas City	#test	www.test.com/postname
19	MM/DD/YYYY	23:59:18	UTC	19	#beta_tester	#test	0	0	0	Arkansas City	#test	www.test.com/postname
20	MM/DD/YYYY	23:58:55	UTC	20	#beta_tester	#test	1	1	1	Arkansas City	#test	www.test.com/postname
21	MM/DD/YYYY	23:58:53	UTC	21	#beta_tester	#test	0	0	0	Arkansas City	#test	www.test.com/postname
22	MM/DD/YYYY	23:58:46	UTC	22	#beta_tester	#test	0	0	0	Arkansas City	#test	www.test.com/postname
23	MM/DD/YYYY	23:58:46	UTC	23	#beta_tester	#test	0	0	0	Arkansas City	#test	www.test.com/postname
24	MM/DD/YYYY	23:58:39	UTC	24	#beta_tester	#test	0	0	0	Arkansas City	#test	www.test.com/postname
25	MM/DD/YYYY	23:58:39	UTC	25	#beta_tester	#test	1	0	0	Arkansas City	#test	www.test.com/postname
26	MM/DD/YYYY	23:58:35	UTC	26	#beta_tester	#test	0	1	2	Arkansas City	#test	www.test.com/postname

Figure 5.1: .csv File Formatting

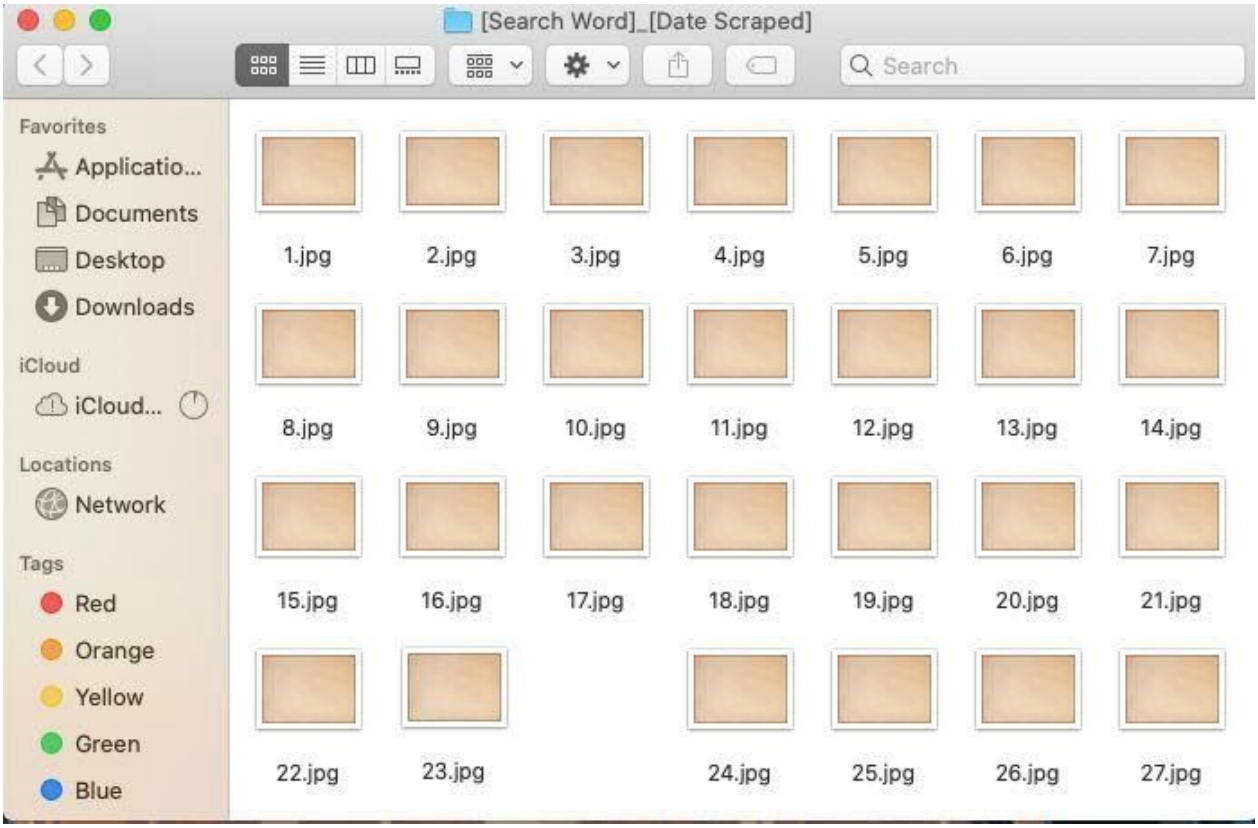
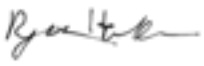
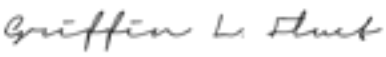
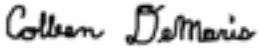


Figure 5.2: Picture File System Formatting

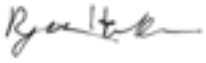
Appendix A - Agreement Between Customer and Contractor

The client (Doctor Rosenbaum) and Blackbear Analytics agree that the system's design meets all requirements prescribed in the SRS document and will be implemented in good faith. All designs will be implemented as close to the specifications in the user interface design document as possible, with margins for error being slim or none. In the event that implementation must diverge from the design, the development team will notify the client as soon as possible and work together with the client until a suitable agreement is reached. If the client would like to add designs/requirements for the system's user interface design after signing this document, Blackbear Analytics bears no responsibility for the failure to implement new designs, but will attempt to add them in good faith. Should any new designs be added, the deliverables schedule in section 2 and 3 of the SRS document will be updated immediately to reflect changes, and the designs/requirements discussed will be reflected in this document by the next deliverable deadline and changes will be finalized with the client.

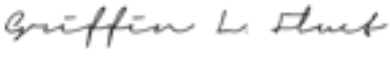
Name	**Signature**	**Date**
Customer:		
Judith E. Rosenbaum		11/25/20
Comments:		
Team:		
Ryan M. Handlon		November 24th, 2020
Comments:		
Abdullah I. Karim		November 24th, 2020
Comments:		
Griffin L. Fluet		November 24th, 2020
Comments:		
Colleen DeMaris		November 24th, 2020
Comments:		
James West		November 24th, 2020

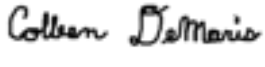
Appendix B - Team Review Sign-off

By signing your name below, you acknowledge that you are a member of Blackbear Analytics and have read the document with an in-depth comprehension of the scribed materials. You agree to complete all designs stated on this document as is, in good faith. Should any new designs come up, you agree to assist in re-drafting this document for approval as described in Appendix A. You agree that it is not required to complete designs added after this version is released, but understand that all designs, new or old, must be implemented in good faith.

Ryan M. Handlon  November 24th, 2020
Comments:

Abdullah I. Karim November 24th, 2020
Comments:

Griffin L. Fluet  November 24th, 2020
Comments:

Colleen DeMaris  November 24th, 2020
Comments:

James West November 24th, 2020

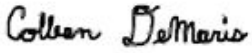
Comments:

Appendix C - Document Contributions

Ryan, Abdullah, Griffin, and Colleen contributed 16.25% total to this document. James contributed 35% because he headed nearly all the design for the mockup on figma. Ryan Handlon created half of the User Interface Walkthrough (Section 3). Abdullah Karim helped in the creation of Data Validation (Section 4) and Report Formats (Section 5). Griffin Fluet contributed massively to the User Interface Standards (Section 2). Colleen DeMaris did half of the User Interface Walkthrough (Section 3). James West created the mockup on figma and helped with the User Interface Standards (Section 2). The Appendices were appended and modified from the SRS document and Appendix C have been read and reviewed by the whole team.

Appendix B – Team Review Sign-off

By signing your name below, you acknowledge that you are a member of Blackbear Analytics and have read the document with an in-depth comprehension of the scribed materials. You agree to complete all designs stated on this document as is, in good faith. Should any new designs come up, you agree to assist in re-drafting this document for approval as described in Appendix A. You agree that it is not required to complete designs added after this version is released, but understand that all designs, new or old, must be implemented in good faith.

Name	Signature	Date
Ryan M. Handlon **Comments:**		December 17th, 2020
Abdullah I. Karim **Comments:**		December 17th, 2020
Griffin L. Fluet **Comments:**		December 17th, 2020
Colleen DeMaris **Comments:**		December 17th, 2020
James West **Comments:**		December 17th, 2020

Appendix C – Document Contributions

Ryan, Abdullah, Griffin, James, and Colleen contributed 20% total to this document. Ryan Handlon created the Table of Contents, List of Figures, List of Tables, Executive Summary, Preface, Summary sections, and imported all Appendices. Abdullah Karim helped in the creation of the Purpose and Method sections. Griffin Fluet and James West contributed to the Design section. Colleen DeMaris created the Testing and Conclusion and Recommendations sections.