

Project name

Software Requirements Specifications Document

~~CSCI 4738, CSCI 4739 Senior Design I and II~~

Project?

Key:

W = Word choice error

C = Clarify, make it specific

Sp = Spelling error

P = punctuation error

M = missing concept or topic

G = grammar error

\ = delete

□ = a graded requirement

Software Requirements Specifications Document

Remove parens *Comparison*
(Social Media Evaluator)
(Social Media Team 1)
(Andres Dicochea, Peter Mansbacher, James Mortensen, Eric
Slick)

Software Requirements Specification

Document

Version: (1.0)

Date: (09/16/2021)

Software Requirements Specifications Document

Table of Contents

1. Introduction	5
1.1 Purpose	5
1.2 Scope	5
1.3 Definitions, Acronyms, and Abbreviations	5
1.4 References	6
1.5 Overview	6
2. The Overall Description	6
2.1 Product Perspective	7
2.1.1 System Interfaces	7
2.1.2 Interfaces	8
2.1.3 Hardware Interfaces	9
2.1.4 Software Interfaces	9
2.1.5 Communications Interfaces	9
2.1.6 Memory Constraints	10
2.1.7 Operations	10
2.1.8 Site Adaptation Requirements	10
2.2 Product Functions	10
2.3 User Characteristics	11
2.4 Constraints	12
2.5 Assumptions and Dependencies	13
2.6 Apportioning of Requirements	13
3. Specific Requirements	13
3.1 External interfaces	14
3.2 Functions	15
3.3 Performance Requirements	16
3.4 Logical Database Requirements	16
3.5 Design Constraints	16
3.5.1 Standards Compliance	16
3.6 Software System Attributes	16
3.6.1 Reliability	17
3.6.2 Availability	17
3.6.3 Security	18
3.6.4 Maintainability	18
3.6.5 Portability	18
3.7 Organizing the Specific Requirements	19
3.7.1 System Mode	19
3.7.2 User Class	19
3.7.3 Objects	19

Software Requirements Specifications Document

3.7.4 Feature	19
3.7.5 Stimulus	19
3.7.6 Response	19
3.7.7 Functional Hierarchy	15
<i>3.8 Additional Comments</i>	<i>19</i>
4. Change Management Process	
5. Document Approvals	
6. Supporting Information	20

Software Requirements Specifications Document

1. Introduction

1.1 Purpose

The purpose of this document is to give a comprehensive overview of the project requirements for "Evaluating Social Media Across Competitors". This SRS document will serve as a reference to the engineers designing and developing the project. It is also a proposal for the client.

1.2 Scope

The problem this project is aiming to solve is can we scrape various social media platforms in order to gather data from posts and feed it to a system that already scores posts for comparison. Capabilities of the software include taking a platform and brand as inputs, and then comparing at least three brands at once. Five or more brands would be better though. Also, the program needs time frames for collecting the posts. This way the comparisons can be graphed against time. The goal is to have an output of, "records in a database for posts that we wanted to get; and a spreadsheet export with the brand, its posts over time and certain data metrics for each post" (Project Description). Benefits will be that with this software, the user will be able to compare the content engagement of companies through different social media platforms. The software being produced for the "Social Media Evaluator" project will be a web interface-based program. The constraints are no API data collection, limited information on posts, legality of having to utilize open-source programs, needing an account, and possibly the collection rate.

1.3 Definitions, Acronyms, and Abbreviations.

Provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendices in the SRS or by reference to documents. This information may be provided by reference to an Appendix.

Table 1- Definitions

Web scrape	Fetching data from various sources to capture structured data elements
GUI	Graphic User Interface
API	Application Programming Interface
Open-Source	Code that is freely available for modification and redistribution
Social Media Platform	A medium for content to be publicized to general public
HTML	Hypertext Markup Language
CSV	Comma Separated Value
Brand Handles	The account name used by brands on social media platforms

Software Requirements Specifications Document

1.4 References

In this subsection:

- (1) Provide a complete list of all documents referenced elsewhere in the SRS
- (2) Identify each document by title, report number (if applicable), date, and publishing organization
- (3) Specify the sources from which the references can be obtained.

This information can be provided by reference to an appendix or to another document. If your application uses specific protocols or RFC's, then reference them here so designers know where to find them.

[1] IEEE 830-1993 - IEEE Recommended Practice for Software Requirements Specifications

[2] Debra Parcheta 9/14/2021 - Social Media Competition- Program to Evaluate Social Media Across Competitors.

1.5 Overview

In this subsection:

- (1) Describe what the rest of the SRS contains
- (2) Explain how the SRS is organized

Don't rehash the table of contents here. Point people to the parts of the document they are most concerned with. Customers/potential users care about section 2, developers care about section 3.

The rest of the SRS contains more detailed information pertaining to the requirements of the project. Customers/potential users care about section two, developers care about sections two and three.

2. The Overall Description

Describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in section 3, and makes them easier to understand. In a sense, this section tells the requirements in plain English for the consumption of the customer. Section 3 will contain a specification written for the developers.

Name it.

Our system will be used to compile information from competing brand's social media. Generally, this system will require the user to specify which social media platform they would like to analyze, the timeframe in which they would like posts to be collected, and the specific brand handles. From there, the client's existing application for analyzing a brand's social media presence will be used to compare multiple brands in the same field. Typically, social media platforms feed information to programs using APIs, which require account login information. Our program will collect all the required information without the login information for specific accounts.

Software Requirements Specifications Document

2.1 Product Perspective

Put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the SRS defines a product that is a component of a larger system, as frequently occurs, then this subsection relates the requirements of the larger system to functionality of the software and identifies interfaces between that system and the software. If you are building a real system, compare its similarity and differences to other systems in the marketplace. If you are doing a research-oriented project, what related research compares to the system you are planning to build.

A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful. This is not a design or architecture picture. It is more to provide context, especially if your system will interact with external actors. The system you are building should be shown as a black box. Let the design document present the internals.

The following subsections describe how the software operates inside various constraints.

Our tool for “Evaluating Social Media Across Competitors” must interact with a number of different services. It will essentially serve as a messenger between social media platforms and the client’s existing social media rating software, relaying all the requested information by the user so that different brand’s social media presences can be compared quantifiably.

Our software will need to interact with the following social media networks:

- Facebook
- Twitter
- Instagram
- TikTok
- YouTube
- Pinterest

From these applications, the following information will be collected for all posts within the user’s designated timeframe:

- Post URL (This URL is unique for every post, and will always get one back to that post even if it is no longer accessible through the platform)
- Post Date
- Post Text
- Image URL (Provides the image location, if applicable to the post)
- Impressions (Number of times the post has been viewed/visited by users of the platform)
- Engagements (Number of times users interacted with the post. The ways in which users can interact with posts varies by platforms, see **2.1.1 System Interfaces** for specific ways interactions are measured for each platform)

This information, along with the user’s designated platform and brand handles, will be entered into our database.

Additionally, our application will interact with the client’s existing social media rating system, so that all the social media posts collected can be compared. This will be done by allowing the client’s system access to our database. Also, all information requested to be scraped by the user will be downloadable via CSV file.

M: You will store data in a database you create.

2.1.1 System Interfaces

List each system interface and identify the functionality of the software to accomplish the system requirement and the interface description to match the system. These are external systems that you have to interact with. For instance, if you are building a business application that interfaces with the existing employee payroll system, what is the API to that system that designer’s will need to use?

Your own footer please

Software Requirements Specifications Document

- Our program will collect the following information each time it is used:
 - Social Media Platform
 - Brand Handle
 - Post URL
 - Post Date
 - Post Text
 - Image URL
 - Impressions
 - Engagements
 - This applies to all aforementioned social media platforms, and it will only collect this information for posts *within* the user's designated timeframe. What qualifies as an engagement may vary depending on the social media platform being scraped. Our program will consider, and retrieve, the following information to measure engagements for each social media platform:
 - Facebook (Likes, Number of Comments, Number of Shares)
 - Twitter (Likes, Number of Comments, Retweets)
 - Instagram (Likes, Number of Comments)
 - TikTok (Likes, Number of Comments, Shares)
 - YouTube (Likes, Number of Comments)
 - Pinterest (Saves, Pins) *Useups?*
 - All information will be collected without the use of APIs.
 - In order for the client's existing rating system to interact with our system, all information will be entered into our database and downloadable via CSV file. From there, the client's system will handle rating each post and analyzing the posts from each brand.
- we're going to need to make a matrix of what we can get for engagements from each platform.*
- oh, here we go
Get this into a chart matrix.*

ODBC connectable

2.1.2 Interfaces

Specify:
(1) The logical characteristics of each interface between the software product and its users.
(2) All the aspects of optimizing the interface with the person who must use the system

This is a description of how the system will interact with its users. Is there a GUI, a command line or some other type of interface? Are there special interface requirements? If you are designing for the general student population for instance, what is the impact of ADA (American with Disabilities Act) on your interface?

- Users will interact with our application via a web interface.
The user-facing web interface will allow the user to specify the following information:
 - The platform they would like the program to scrape (The user will only be able to scrape one platform at a time)
 - This selection can be made by the user clicking on an image of the platform's logo *mece.*
 - The timeframe in which the user would like posts from (This timeframe cannot exceed 90 days) *max.*
 - The user will input in the format: YYYY/MM/DD *no, "handle it"*
 - The brand handles the user would like to retrieve posts from (maximum of 3)

M: minimum time frame?

*uri?
S?*

Software Requirements Specifications Document

- Based on the selected platform, a message will appear above the inputs for brand handles which will explain to the user how to find the handle on that specific platform
← facilitate too?
- After the data is collected by our program, the web interface will be updated with a button labeled “Download as CSV file”.

2.1.3 Hardware Interfaces

Specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics. It also covers such matters as what devices are to be supported, how they are to be supported and protocols. This is not a description of hardware requirements in the sense that “This program must run on a Mac with 64M of RAM”. This section is for detailing the actual hardware devices your application will interact with and control. For instance, if you are controlling X10 type home devices, what is the interface to those devices? Designers should be able to look at this and know what hardware they need to worry about in the design. Many business type applications will have no hardware interfaces. If none, just state “The system has no hardware interface requirements”. If you just delete sections that are not applicable, then readers do not know if: a. this does not apply or b. you forgot to include the section in the first place.

The hardware interfaces for this project are minimal. As it is a web-based application, it will be interacting with any devices capable of accessing a web page, primarily desktop computers and tablets. Our program does not require any input from hardware interfaces for data collection.

2.1.4 Software Interfaces

Specify the use of other required software products and interfaces with other application systems. For each required software product, include:

- (1) Name
- (2) Mnemonic
- (3) Specification number
- (4) Version number
- (5) Source

M: Browsers

Commented [MJ1]: add social media platforms, python and BS versions

For each interface, provide:
(1) Discussion of the purpose of the interfacing software as related to this software product
(2) Definition of the interface in terms of message content and format

Here we document the APIs, versions of software that we do not have to write, but that our system has to use. For instance if your customer uses SQL Server 7 and you are required to use that, then you need to specify i.e.
2.1.4.1 Microsoft SQL Server 7. The system must use SQL Server as its database component. Communication with the DB is through ODBC connections. The system must provide SQL data table definitions to be provided to the company DBA for setup.

A key point to remember is that you do NOT want to specify software here that you think would be good to use. This is only for customer-specific systems that you have to interact with. Choosing SQL Server 7 as a DB without a customer requirement is a Design choice, not a requirement. This is a subtle but important point to writing good requirements and not over-constraining the design.

software system name

Python 3 will be used to design our web scraper. Our system will receive an HTML page from the most recent version of each social media platform and parse the required information.

- Communication between our system and each social media platform will be performed by our system based on the BeautifulSoup framework, which will retrieve any specified HTML pages.

reference the git and doc
M: Selenium ?

2.1.5 Communications Interfaces

We
don't
choose
technologies
in this document.

Software Requirements Specifications Document

Specify the various interfaces to communications such as local network protocols, etc. These are protocols you will need to directly interact with. If you happen to use web services transparently to your application then do not list it here. If you are using a custom protocol to communicate between systems, then document that protocol here so designers know what to design. If it is a standard protocol, you can reference an existing document or RFC.

- Standard TCP/IP communication will be used by our web host to communicate with the user [RFC 793].
- No custom communication interfaces are necessary.

2.1.6 Memory Constraints

Specify any applicable characteristics and limits on primary and secondary memory. Don't just make up something here. If all the customer's machines have only 128K of RAM, then your target design has got to come in under 128K so there is an actual requirement. You could also cite market research here for shrink-wrap type applications "Focus groups have determined that our target market has between 256-512M of RAM, therefore the design footprint should not exceed 256M." If there are no memory constraints, so state.

There are no memory constraints.

what do we need at our server?
Processor speed?
Storage?

2.1.7 Operations

Specify the normal and special operations required by the user such as:

- (1) The various modes of operations in the user organization
- (2) Periods of interactive operations and periods of unattended operations
- (3) Data processing support functions
- (4) Backup and recovery operations

(Note: This is sometimes specified as part of the User Interfaces section.) If you separate this from the UI stuff earlier, then cover business process type stuff that would impact the design. For instance, if the company brings all their systems down at midnight for data backup that might impact the design. These are all the work tasks that impact the design of an application, but which might not be located in software.

There are no operational requirements.

- you need a server

2.1.8 Site Adaptation Requirements

In this section:

- (1) Define the requirements for any data or initialization sequences that are specific to a given site, mission, or operational mode
- (2) Specify the site or mission-related features that should be modified to adapt the software to a particular installation

If any modifications to the customer's work area would be required by your system, then document that here. For instance, "A 100Kw backup generator and 10000 BTU air conditioning system must be installed at the user site prior to software installation". This could also be software-specific like, "New data tables created for this system must be installed on the company's existing DB server and populated prior to system activation." Any equipment the customer would need to buy or any software setup that needs to be done so that your system will install and operate correctly should be documented here.

M: internet connection

As our project is entirely web-based, there are likely no site adaptation requirements, unless the user does not have a device which can access a web page.

2.2 Product Functions

Provide a summary of the major functions that the software will perform. Sometimes the function summary that is necessary for this part can be taken directly from the section of the higher-level specification (if one exists) that allocates particular functions to the software product.

For clarity:

Software Requirements Specifications Document

- (1) The functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time.
- (2) Textual or graphic methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product but simply shows the logical relationships among variables.

AH, Finally the real meat of section 2. This describes the functionality of the system in the language of the customer. What specifically does the system that will be designed have to do? Drawings are good, but remember this is a description of what the system needs to do, not how you are going to build it. (That comes in the design document).

The product functions have briefly been described prior to this section of the SRS document. This is a more in-depth summary of what specifically the system does. Software for the project will create a web-based user interface. The "Social Media Evaluator" product is a system with a user interface that allows the user to pick several brand handles at a time and compare their user engagement based on the user's specified social media platform. Then the user will have to define a start and end date for the time frame (formatted as YYYY/MM/DD). Once the brand handles and time frame have been defined, the system will search the specified platform for each brand. Then, the following data is required by the client to be scraped:

- Platform Name (text Facebook, TikTok, Instagram, YouTube, Pinterest, Twitter - could be parsed from the url)
- Brand handle (text, the account name used by the brand we want to search for on the platform)
- Post URL (text, some platforms call this the permalink or the link, it's the url that always gets you back to this particular post even if it's no longer showing on the platform because of age or censure)
- Post Date (date, a timestamp is also fine, it can be parsed)
- Post Text (Whatever the message was in the post, some messages have no text....)
- Impressions (the number of times the post has been viewed - not unique)
- Engagements (a number, a count of the actions that consumers do when they see the post, varies by platform)

In order to scrape the data, the development team will have to use frameworks and not the social media platforms' own API. The data parsed will be put into a separate system to hold the data and feed the data into a database system. Once complete, the interface will show the user the list of posts and give the option to download to a spreadsheet or csv file. The user won't have access to the final part of the project where the csv file is fed into the clients' system. The clients' system will take the organized, scraped data and use it to create a scatterplot visualizing impressions over time.

M: show a comparison of the brand handles used.

2.3 User Characteristics

Describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. Do not state specific requirements

Software Requirements Specifications Document

but rather provide the reasons why certain specific requirements are later specified in section 3.

What is it about your potential user base that will impact the design? Their experience and comfort with technology will drive UI design. Other characteristics might actually influence internal design of the system.

As described by the client, the users of the product are non-computer science personnel. These individuals will most likely not have any high-level technical skills for computing and will need a simple interface for inputting and receiving output. These specifications will impact the design by making sure developers keep complex, confusing information behind the scenes so the user can get their results without any unnecessary complications.

2.4 Constraints

Provide a general description of any other items that will limit the developer's options. These can include:

- (1) Regulatory policies
- (2) Hardware limitations (for example, signal timing requirements)
- (3) Interface to other applications
- (4) Parallel operation
- (5) Audit functions
- (6) Control functions
- (7) Higher-order language requirements
- (8) Signal handshake protocols (for example, XON-XOFF, ACK-NACK)
- (9) Reliability requirements
- (10) Criticality of the application
- (11) Safety and security considerations

This section captures non-functional requirements in the customer's language. A more formal presentation of these will occur in section 3.

1. The "Social Media Evaluator" applied is constrained by the developers not being able to use the social media platforms API. This may cause the developers to run into certain blocks put in place by the social media platforms.
2. Each site has different roadblocks put in place. Figuring out how to bypass them for each social media platform will take different amounts of time most likely. Some will be easier than others.
3. Only three brands are required to be able to be compared, with 5 or more being better. Sometimes businesses may want to compare many brands at once and this would be a constraint. The more brands being searched, the longer it will take the program.
4. For this project, developers are only going to be scraping public found data on the social media platforms.
5. Social media platforms create situations where developers need to throttle while scraping in order to bypass the platforms connection blocks.

Software Requirements Specifications Document

2.5 Assumptions and Dependencies

List each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system would be available on the hardware designated for the software product. If, in fact, the operating system were not available, the SRS would then have to change accordingly.

This section is catch-all for everything else that might influence the design of the system and that did not fit in any of the categories above.

One assumption being made is that this application will be run on a device with internet connection. This application relies on having the internet to access the data to be scraped and to access the database that will hold data.

*M:
browser
minimum?
version?*
Another assumption being made is that the users computing device will have the similarly up-to-date technology (operating system, web browser(s)) to be compatible with the web-scraping technology that the developers of this project will implement.

Something that will be important with future social media platforms is portability. The developers of this project will either do each platform by hand or code will be portable between platforms; if portable between the assigned, it is assumed that future platforms will be able to be scraped similarly.

Talk about constructor design.

2.6 Apportioning of Requirements.

Identify requirements that may be delayed until future versions of the system. After you look at the project plan and hours available, you may realize that you just cannot get everything done. This section divides the requirements into different sections for development and delivery. Remember to check with the customer – they should prioritize the requirements and decide what does and does not get done. This can also be useful if you are using an iterative life cycle model to specify which requirements will map to which iteration.

Never

One future version of the system will be a stand-alone, web-based application. In the future, it is also hoped that the application will be able to scrape from every client-mentioned social media platform as well as others not listed, already existing currently or in the future.

CH 3. Specific Requirements

Software Requirements Specifications Document

(- and you should not ...)

As of 9/27/2021 we currently do not have the exact system that this application will run on. From our last discussion with the client, Debra Parcheta, the only details we received was that the user would log on through a desktop. Debra is checking on the system that the database can be built which we maybe a school server.

3.1.1) User Interface shall contain 6 different social media platforms that the user can select to compare each of the products. The following social media platforms will be an option for the user to scrape:

- Facebook
- TikTok
- Instagram
- Pinterest
- YouTube
- Twitter

The user will have the option to choose one platform per **W**.

3.1.2) User Interface will contain 5 input fields

- 3.1.2.1) 3 input fields for brand handles
- 3.1.2.2) 2 input fields for date ranges

3.1.4) The output of the software will produce a list containing:

- Platform Name
- Brand handle
- Post URL
- Post Date
- Post Text
- Impressions
- Engagements

- C

This information will be uploaded to the client's database.

3.1.5) The software will contain **an** option for user to download a CSV file to their **device** system. This CSV file will consist of information stated in **3.1.4** from the most recent search query that the user inputted. Q

3.1 External Interfaces

- Search Interface

- **Purpose:** This interface will be presented to user once they successfully access the web page Interface. The purpose of this will be for the user to enter up to 3 brand handles that the client is searching for and update a database with all information that was scraped.
- **Input:** The user will be asked to enter 5 fields for beginning a search query

Software Requirements Specifications Document

- Up to 3 brand handles
- 2 date fields (start – end)
- **Output:**
 - The destination of the output will be the system that is supplied by the client *rewrite M: onscreen, csv*
- **Valid Input:**
 - Brand handles: The brand handles need to be a perfect match to what the social media page is registered as. If there are any differences between the input and the actual brand handle, the user will receive an error message that could not be found.
 - Date ranges: The date ranges that are input need to be formatted in a chronological order, in which the start date cannot be a future date compared to the end date. *M: error?*
- **Relationships to other inputs/outputs:**
 - The web scraping query that the user enters in the website interface will be an input for the database *C, you'll store the query?*
- **Screen Formats:**
 - Different screen formats are being considered and will be shown to clients in near future. Once an agreement is reached, the official screen format will be presented here.
- **Data Formats:**
 - Brand handles: minimum of 1 character long. Format will be dependent on social media platform
 - Date: YYYY/MM/DD *handle it?*
- **End messages:** N/A

3.2 Functions

3.2.1) Validity Checks:

- The system shall check for correct handles which are input for the search queries
- The system shall check for correct dates that are inputted and are formatted correctly (YYYY/MM/DD) *- handling*

3.2.2) Exact Sequence of Operations:

- 3.2.2.1)** The user will input three handles and a date range into the GUI and the application shall check if the input is valid. If the search query is valid, then the application will update the client's database with information that was collected. A validation message will be presented to the user in which that the search query was successful. If the search query was not successful, then the application will display a proper error message to the user and ask them to try again.

3.2.3) Responses to Abnormal situations

- Incorrect brand handle:
 - Message printed to user that the brand handle field is incorrect and to check if it is correct
- Incorrect date format:

Software Requirements Specifications Document

- Message printed to user that the date fields are incorrect and to check if input as YYYY/MM/DD handle
- Communication to remote database failure:
 - The system shall print a message to the user that the system database cannot be communicated. The user will be asked to contact the database administrators.

3.3 Performance Requirements

- Revise*
- A) The current number of terminals that will be supported at this time will be 1. This will be able to scale for more users in the future and if requirements change for the client.
B) At this current time, there will only be one user that will be using our system at a given time.
C) The amount of information that will be handled will have a wide range of storage dependent on the amount of search queries that are performed in a specific time frame. The database will store search queries for only 90 days. The storage constraints will be dependent on the system that the client provides.
- immediate*
- not ok*
- longer*
- what's needed?*

Dynamic numerical requirements:

- 100% of the data scraping will be performed within 24 hours of the user input
- The amount of brand handles that are entered and date range will be significant variables for the search query to complete

3.4 Logical Database Requirements

- no, they won't, no relationship.*
- Storing queries?*
- Data retention requirements will be that the database will hold a search query for up to the client
 - Data entities will have up to three relationships depending on the search query that the user enters. These relationships will be between the different brand handles that were initially entered.

3.5 Design Constraints

The design constraints that will be presented are dependent on the system that is supplied for our database. As of 9/28/2021, we are currently waiting for the database system that we will have access to use.

No - get specific - your design

3.5.1 Standards Compliance

Revise

As of 9/28/2021 we currently do not have any existing standards or regulations.

3.6 Software System Attributes

Software Requirements Specifications Document

- The website will contain a web-scraping tool that can scrape the following 6 social media outlets:
 - Twitter, Pinterest, Instagram, Facebook, TikTok, Youtube
- The website will communicate with a database and store scraped logs with relevant information such as date and the brand name that was scraped

what?
we're -?
logging-

3.6.1 Reliability

Website:

- The website will be responsible for evaluating the validity of returned scraping data for accuracy and error messages

rewrite

Database:

- The database will be evaluated each day for uptime in order to validate the correct storage of scraped information

C?

3.6.2 Availability

Website:

- Maintain session information relating to:

- Data inputs
 - Desired brand to be scraped
 - Desired social media outlets to be scraped
 - Time frame of data to pull
- User login

The website will be expected to recover from any crashing or downtime within an hour.

C?

- Upon crashing of website

- Identify data loss associated with previously queried information whether from database or prior to web-scraping

- Notify of downtime

- 24 hours ahead of scheduled maintenance
- 1 hour before scheduled maintenance
- The website will be expected to restart daily and must adhere to previous requirements of notifying users ahead of scheduled downtime except for the 24-hour requirement due to frequency of restarts.

{ your software does this? }

Why?

- Daily restart

- Occurring at midnight each night
- Notifying users 1 hour ahead of time 11am or according to timezone
- Notifications must account for timezone differences across userbase

?C

- Requested brand information

- Requested time frame

- Requested social media outlets

?C

The web-scraping tool will recover given any bad data inputs based on the failed request

?C

- Notify user of invalid URL

?C

- Notify user of failure to obtain requested information

?C

- Notify user if information does not exist
- Notify user if information cannot be scraped

Software Requirements Specifications Document

Database:

The database will checkpoint each night upon restart of the website and back-up the previously queried information.

- Information regarding brands will be backed-up in case of loss of data or database functionality for an extended period
 - Brand name, posts and timeframes

The database will recover given any number of potential failures including but not limited to power loss. *- your software does this?*

The database will restart daily at midnight MST in line with the website to maintain functionality. *what downtime? Why?*

- Notify users of downtime 1 hour ahead of time with website notification
- Will not notify users 24 hours ahead of time due to frequency of restarts

3.6.3 Security

Website:

Website front-end will be entirely separated from communication with the database

Website backend will perform security checks related to webscraped data in order to avoid critical vulnerabilities and failures in the database

- Input Validation
- Requested information is valid IE not "404 Error" being stored

Website will utilize SSO and integrate Google login or password hashes if stored locally/in the database *overkill*

Website will utilize input validation to avoid SQL injection, cross-site scripting, etc *Not needed though?*

Database:

Database will keep log and history of previously queried/scraped results and the user who requested them

3.6.4 Maintainability

~~Someone else will be maintaining the system~~ *- o*

Website:

Website will have a simple interface for user input queries regarding brand, timeframe, social media outlets to be scraped.

~~Existing frameworks including BS4 and Python's~~ will be updated accordingly *Don't choose tech here*

Daily tests to test working functionality of the web scraper

Database:

Validate that database is storing data correctly and that previously scraped data still exists in the database as long as it is not passed the expired date

3.6.5 Portability

Software Requirements Specifications Document

No - you write the code

- * 80% of code will be host dependent, based on Python and running on the server.
- * Python frameworks BS4, Selenium, etc will be used. Given nature of interpreted language, can be run on any environment with Python3.
- * Python interpreter, subsequent dependencies, BS4, html.parser, etc.
- * No particular Operating System is required, Python is an interpreted language and therefore all the hosting server will need is Python and its dependencies.

3.7 Organizing the Specific Requirements

3.7.1 System Mode

look up this term.

Interfaces will be website based and functionality will be back-end based still tied to the website.

3.7.2 User Class

The only class of users will be those who are querying the webscraper for data about a brand.

M: Admin user

3.7.3 Objects

Brand Class which Brand Children can inherit

- Brand will be related to the real world brand and subsequently will have corresponding web-scraping functionality for each social media outlet

3.7.4 Feature

Web-Scraping will be based on user input of:

- Brand name
- Date range

needs much more C

3.7.5 Stimulus

I don't believe this is relevant to the project.

look it up

it is.

3.7.6 Response

look it up.

Expected response from social media outlet containing scraped information related to engagement and

3.8 Additional Comments

Software Requirements Specifications Document

No additional comments as of current state of project.

Formatting
bold?

4. Change Management Process

Still identifying the change management process. Will be run accessed and maintained via Github for coding changes.

why bold?
Change log process
must be used.

5. Document Approvals

The potential people approving of the SRS document will be listed below:

- Andres Dicochea
- James Mortensen
- Eric Slick
- Peter Mansbacher

M: Client

The team will be in a 2-week rotation to switch who the team leader will be. The current team leader will be the approver of the SRS document and will sign and date the document

all

6. Supporting Information

The supporting information makes the SRS easier to use. It includes:

- * Table of Contents
- * Index
- * Appendices

The Appendices are not always considered part of the actual requirements specification and are not always necessary. They may include:

- (a) Sample I/O formats, descriptions of cost analysis studies, results of user surveys
- (b) Supporting or background information that can help the readers of the SRS
- (c) A description of the problems to be solved by the software
- (d) Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements

When Appendices are included, the SRS should explicitly state whether or not the Appendices are to be considered part of the requirements.

Tables on the following pages provide alternate ways to structure section 3 on the specific requirements. You should pick the best one of these to organize section 3 requirements.

Software Requirements Specifications Document

Outline for SRS Section 3 Organized by mode: Version 1

- 3. Specific Requirements
 - 3.1 External interface requirements
 - 3.1.1 User interfaces
 - 3.1.2 Hardware interfaces
 - 3.1.3 Software interfaces
 - 3.1.4 Communications interfaces
 - 3.2 Functional requirements
 - 3.2.1 Mode 1
 - 3.2.1.1 Functional requirement 1.1
 -
 - 3.2.1.*n* Functional requirement 1.*n*
 - 3.2.2 Mode 2
 -
 - 3.2.*m* Mode *m*
 - 3.2.*m*.1 Functional requirement *m*.1
 -
 - 3.2.*m*.*n* Functional requirement *m*.*n*
 - 3.3 Performance Requirements
 - 3.4 Design Constraints
 - 3.5 Software system attributes
 - 3.6 Other requirements

Software Requirements Specifications Document

Outline for SRS Section 3 Organized by mode: Version 2

- 3. Specific Requirements
 - 3.1 Functional Requirements
 - 3.1.1 Mode 1
 - 3.1.1.1 External interfaces
 - 3.1.1.1 User interfaces
 - 3.1.1.2 Hardware interfaces
 - 3.1.1.3 Software interfaces
 - 3.1.1.4 Communications interfaces
 - 3.1.1.2 Functional Requirement
 - 3.1.1.2.1 Functional requirement 1
 -
 - 3.1.1.2.n Functional requirement n
 - 3.1.1.3 Performance
 - 3.1.2 Mode 2
 -
 - 3.1.m Mode m
 - 3.2 Design constraints
 - 3.3 Software system attributes
 - 3.4 Other requirements

Software Requirements Specifications Document

Outline for SRS Section 3

Organized by user class (i.e. different types of users ->System Administrators, Managers, Clerks, etc.)

- 3. Specific Requirements
 - 3.1 External interface requirements
 - 3.1.1 User interfaces
 - 3.1.2 Hardware interfaces
 - 3.1.3 Software interfaces
 - 3.1.4 Communications interfaces
 - 3.2 Functional requirements
 - 3.2.1 User class 1
 - 3.2.1.1 Functional requirement 1.1
 -
 - 3.2.1.n Functional requirement 1.n
 - 3.2.2 User class 2
 -
 - 3.2.m User class m
 - 3.2.m.1 Functional requirement m.1
 -
 - 3.2.m.n Functional requirement m.n
 - 3.3 Performance Requirements
 - 3.4 Design Constraints
 - 3.5 Software system attributes
 - 3.6 Other requirements

Software Requirements Specifications Document

Outline for SRS Section 3

Organized by object (Good if you did an object-oriented analysis as part of your requirements)

3 Specific Requirements

3.1 External interface requirements

 3.1.1 User interfaces

 3.1.2 Hardware interfaces

 3.1.3 Software interfaces

 3.1.4 Communications interfaces

3.2 Classes/Objects

 3.2.1 Class/Object 1

 3.2.1.1 Attributes (direct or inherited)

 3.2.1.1.1 Attribute 1

 3.2.1.1.n Attribute n

 3.2.1.2 Functions (services, methods, direct or inherited)

 3.2.1.2.1 Functional requirement 1.1

 3.2.1.2.m Functional requirement 1.m

 3.2.1.3 Messages (communications received or sent)

 3.2.2 Class/Object 2

 3.2.p Class/Object p

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

Software Requirements Specifications Document

Outline for SRS Section 3 **Organized by feature (Good when there are clearly delimited feature sets.**

- 3 Specific Requirements
 - 3.1 External interface requirements
 - 3.1.1 User interfaces
 - 3.1.2 Hardware interfaces
 - 3.1.3 Software interfaces
 - 3.1.4 Communications interfaces
 - 3.2 System features
 - 3.2.1 System Feature 1
 - 3.2.1.1 Introduction/Purpose of feature
 - 3.2.1.2 Stimulus/Response sequence
 - 3.2.1.3 Associated functional requirements
 - 3.2.1.3.1 Functional requirement 1
 -
 - 3.2.1.3.*n* Functional requirement *n*
 - 3.2.2 System Feature 2
 -
 - 3.2.*m* System Feature *m*
 -
 - 3.3 Performance Requirements
 - 3.4 Design Constraints
 - 3.5 Software system attributes
 - 3.6 Other requirements

Software Requirements Specifications Document

Outline for SRS Section 3

Organized by stimulus (Good for event driven systems where the events form logical groupings)

- 3 Specific Requirements
 - 3.1 External interface requirements
 - 3.1.1 User interfaces
 - 3.1.2 Hardware interfaces
 - 3.1.3 Software interfaces
 - 3.1.4 Communications interfaces
 - 3.2 Functional requirements
 - 3.2.1 Stimulus 1
 - 3.2.1.1 Functional requirement 1.1
 -
 - 3.2.1.n Functional requirement 1.n
 - 3.2.2 Stimulus 2
 -
 - 3.2.m Stimulus m
 - 3.2.m.1 Functional requirement m.1
 -
 - 3.2.r.n Functional requirement m.n
 - 3.3 Performance Requirements
 - 3.4 Design Constraints
 - 3.5 Software system attributes
 - 3.6 Other requirements

Software Requirements Specifications Document

Outline for SRS Section 3

Organized by response (Good for event driven systems where the responses form logical groupings)

- 3 Specific Requirements
 - 3.1 External interface requirements
 - 3.1.1 User interfaces
 - 3.1.2 Hardware interfaces
 - 3.1.3 Software interfaces
 - 3.1.4 Communications interfaces
 - 3.2 Functional requirements
 - 3.2.1 Response 1
 - 3.2.1.1 Functional requirement 1.1
 -
 - 3.2.1.*n* Functional requirement 1.*n*
 - 3.2.2 Response 2
 -
 - 3.2.*m* Response *m*
 - 3.2.*m*.1 Functional requirement *m*.1
 -
 - 3.2.*m*.*n* Functional requirement *m*.*n*
 - 3.3 Performance Requirements
 - 3.4 Design Constraints
 - 3.5 Software system attributes
 - 3.6 Other requirements

Software Requirements Specifications Document

Outline for SRS Section 3

Organized by functional hierarchy (Good if you have done structured analysis as part of your design.)

- 3 Specific Requirements
 - 3.1 External interface requirements
 - 3.1.1 User interfaces
 - 3.1.2 Hardware interfaces
 - 3.1.3 Software interfaces
 - 3.1.4 Communications interfaces
 - 3.2 Functional requirements
 - 3.2.1 Information flows
 - 3.2.1.1 Data flow diagram 1
 - 3.2.1.1.1 Data entities
 - 3.2.1.1.2 Pertinent processes
 - 3.2.1.1.3 Topology
 - 3.2.1.2 Data flow diagram 2
 - 3.2.1.2.1 Data entities
 - 3.2.1.2.2 Pertinent processes
 - 3.2.1.2.3 Topology
 -
 - 3.2.1.n Data flow diagram n
 - 3.2.1.n.1 Data entities
 - 3.2.1.n.2 Pertinent processes
 - 3.2.1.n.3 Topology
 - 3.2.2 Process descriptions
 - 3.2.2.1 Process 1
 - 3.2.2.1.1 Input data entities
 - 3.2.2.1.2 Algorithm or formula of process
 - 3.2.2.1.3 Affected data entities
 - 3.2.2.2 Process 2
 - 3.2.2.2.1 Input data entities
 - 3.2.2.2.2 Algorithm or formula of process
 - 3.2.2.2.3 Affected data entities
 -
 - 3.2.2.m Process m
 - 3.2.2.m.1 Input data entities
 - 3.2.2.m.2 Algorithm or formula of process
 - 3.2.2.m.3 Affected data entities
 - 3.2.3 Data construct specifications
 - 3.2.3.1 Construct 1
 - 3.2.3.1.1 Record type
 - 3.2.3.1.2 Constituent fields
 - 3.2.3.2 Construct 2
 - 3.2.3.2.1 Record type
 - 3.2.3.2.2 Constituent fields
 -

Software Requirements Specifications Document

- 3.2.3.*p* Construct *p*
 - 3.2.3.*p*.1 Record type
 - 3.2.3.*p*.2 Constituent fields
- 3.2.4 Data dictionary
 - 3.2.4.1 Data element 1
 - 3.2.4.1.1 Name
 - 3.2.4.1.2 Representation
 - 3.2.4.1.3 Units/Format
 - 3.2.4.1.4 Precision/Accuracy
 - 3.2.4.1.5 Range
 - 3.2.4.2 Data element 2
 - 3.2.4.2.1 Name
 - 3.2.4.2.2 Representation
 - 3.2.4.2.3 Units/Format
 - 3.2.4.2.4 Precision/Accuracy
 - 3.2.4.2.5 Range
 -
 - 3.2.4.*q* Data element *q*
 - 3.2.4.*q*.1 Name
 - 3.2.4.*q*.2 Representation
 - 3.2.4.*q*.3 Units/Format
 - 3.2.4.*q*.4 Precision/Accuracy
 - 3.2.4.*q*.5 Range
- 3.3 Performance Requirements
- 3.4 Design Constraints
- 3.5 Software system attributes
- 3.6 Other requirements

Software Requirements Specifications Document

Outline for SRS Section 3 Showing multiple organizations (Can't decide? Then glob it all together)

- 3 Specific Requirements
 - 3.1 External interface requirements
 - 3.1.1 User interfaces
 - 3.1.2 Hardware interfaces
 - 3.1.3 Software interfaces
 - 3.1.4 Communications interfaces
 - 3.2 Functional requirements
 - 3.2.1 User class 1
 - 3.2.1.1 Feature 1.1
 - 3.2.1.1.1 Introduction/Purpose of feature
 - 3.2.1.1.2 Stimulus/Response sequence
 - 3.2.1.1.3 Associated functional requirements
 - 3.2.1.2 Feature 1.2
 - 3.2.1.2.1 Introduction/Purpose of feature
 - 3.2.1.2.2 Stimulus/Response sequence
 - 3.2.1.2.3 Associated functional requirements
 -
 - 3.2.1.m Feature 1.m
 - 3.2.1.m.1 Introduction/Purpose of feature
 - 3.2.1.m.2 Stimulus/Response sequence
 - 3.2.1.m.3 Associated functional requirements
 - 3.2.2 User class 2
 -
 - 3.2.n User class n
 - 3.3 Performance Requirements
 - 3.4 Design Constraints
 - 3.5 Software system attributes
 - 3.6 Other requirements

Software Requirements Specifications Document

Outline for SRS Section 3

Organized by Use Case (Good when following UML development)

3. Specific Requirements

3.1 External Actor Descriptions

3.1.1 Human Actors

3.1.2 Hardware Actors

3.1.3 Software System Actors

3.2 Use Case Descriptions

3.2.1 Use Case 1

3.2.2 Use Case 2

3.2.n Use Case n

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements