UML DIAGRAMS ON WEB SCRAPING

April 29, 2023

1 UML Introduction

A software development method consists of a modeling language and a process. The Unified Modeling Language (UML) is called a modeling language, not a method. The modeling language is the notation that methods use to express designs. The process describes the steps taken in doing a design. The Unified Modeling Language (UML) is developed as a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. In the UML, the five main views of the system are

2 Types of UML Diagram

So what are the different UML diagram types? There are two main categories; structure diagrams and behavioral diagrams. Click on the links to learn more about a specific diagram type.

Structure Diagrams

- class Diagram
- Component Diagram
- Deployment Diagram
- Object Diagram
- Package Diagram
- Profile Diagram
- Composite Structure Diagram

Behavioral Diagrams

- Use Case Diagram
- Activity Diagram
- State Machine Diagram
- Sequence Diagram
- Communication Diagram
- Interaction Overview Diagram
- Timing Diagram

1.USECASE DIAGRAM

AIM : To Draw A Usecase Diagram For WebScraping

DESCRIPTION : A use case diagram for web scraping typically involves two actors: the user and the web scraper.

The user is typically an individual or an organization that wants to extract data from one or more websites. The user initiates the process by providing the URLs of the target websites and specifying the data to be extracted.

The web scraper is a software program that runs on a server or a personal computer and performs the actual web scraping. The web scraper reads the HTML code of the target website, identifies the relevant data, and extracts it according to the user's specifications.

The following are the main use cases in a typical web scraping use case diagram:

1. Start Web Scraping:The user starts the web scraping process by providing the URLs of the target websites and specifying the data to be extracted.
2. Authenticate User: The web scraper authenticates the user by verifying their credentials.
3. Identify Data: The web scraper reads the HTML code of the target website and identifies the relevant data.
4. Extract Data: The weStop Web Scraping: The user stops the web scraping process if it is no longer required.b scraper extracts the data from the target website according to the user's specifications.
5. Store Data: The web scraper stores the extracted data in a database or a file for later use.
6. Send Notification: The web scraper sends a notification to the user when the web scraping process is completed.
7. Stop Web Scraping: The user stops the web scraping process if it is no longer required.

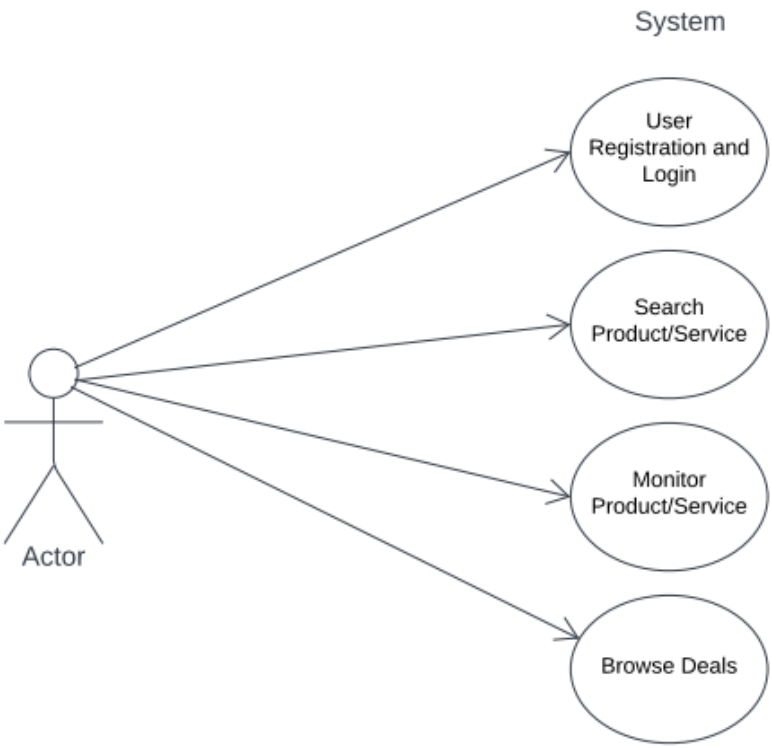

Figure 1: USECASE DIAGRAM

2.ACTIVITY DIAGRAM

AIM : To Draw A Activity Diagram For Web Scraping

Description : An activity diagram for web scraping is a graphical representation of the steps involved in the process of extracting data from one or more websites. It describes the flow of activities, decisions, and actions involved in the web scraping process. Here is a description of the main elements of an activity diagram for web scraping:

1. Start: The start node represents the beginning of the web scraping process.
2. Provide URL: The user provides the URL(s) of the target website(s) to the web scraper.
3. Authenticate User: The web scraper authenticates the user by verifying their credentials.
4. Read HTML Code: The web scraper reads the HTML code of the target website to identify the relevant data.
5. Identify Data: The web scraper identifies the data to be extracted according to the user's specifications.
6. Extract Data: The web scraper extracts the data from the target website according to the user's specifications.
7. Store Data: The web scraper stores the extracted data in a database or a file for later use.
8. Check for More URLs: If the user has provided multiple URLs, the web scraper checks for additional URLs.
9. Stop Web Scraping: The user stops the web scraping process if it is no longer required.
10. Send Notification: The web scraper sends a notification to the user when the web scraping process is completed.
11. End: The end node represents the completion of the web scraping process.

The activity diagram would show these steps as a series of connected actions or nodes, with arrows indicating the flow of data between them. It might also include decision points, such as whether to continue scraping additional pages, based on the results of a previous extraction. The overall goal of the activity diagram is to provide a clear, visual representation of the web scraping process, from start to finish.

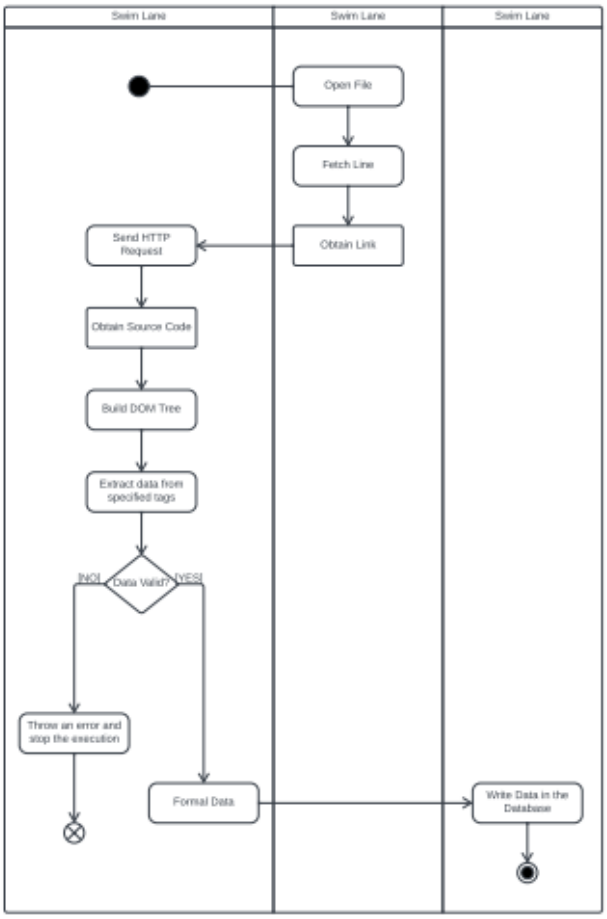

Figure 2: ACTIVITY DIAGRAM

3.SEQUENCE DIAGRAM

AIM : To Draw A Sequence Diagram For Web Scraping

Description : A sequence diagram for web scraping is a visual representation of the interactions between the different components involved in the web scraping process. It shows the sequence of messages exchanged between the user and the web scraper, and the order in which they occur. Here is a description of the main elements of a sequence diagram for web scraping:

1. User: The user initiates the web scraping process by providing the URL(s) of the target website(s) to the web scraper.
2. Web Scraper: The web scraper receives the URL(s) from the user and sends an authentication request to verify the user's credentials.
3. User: The user responds to the authentication request by providing their credentials.
4. Web Scraper: The web scraper verifies the user's credentials and sends a request to the target website(s) to retrieve the HTML code.
5. Target Website: The target website receives the request from the web scraper and sends back the HTML code.
6. Web Scraper: The web scraper receives the HTML code and parses it to identify the relevant data.
7. Web Scraper: The web scraper extracts the relevant data from the HTML code and stores it in a database or a file.
8. User: The user may request the web scraper to extract additional data from the same or different websites by providing more URLs.
9. Web Scraper: The web scraper repeats the steps from 2 to 7 for each additional URL provided by the user.
10. Web Scraper: Once all the requested data has been extracted and stored, the web scraper sends a notification to the user indicating that the web scraping process is complete.

The sequence diagram would typically begin with a request from the user to scrape data from a particular website. This request would be passed to the web scraping tool, which would then send a request to the website's server to access the desired web page.

Once the web page is accessed, the scraping tool would then parse the HTML code of the web page to extract the relevant data. This might involve using regular expressions, HTML tags, or other techniques to identify and extract the desired data.

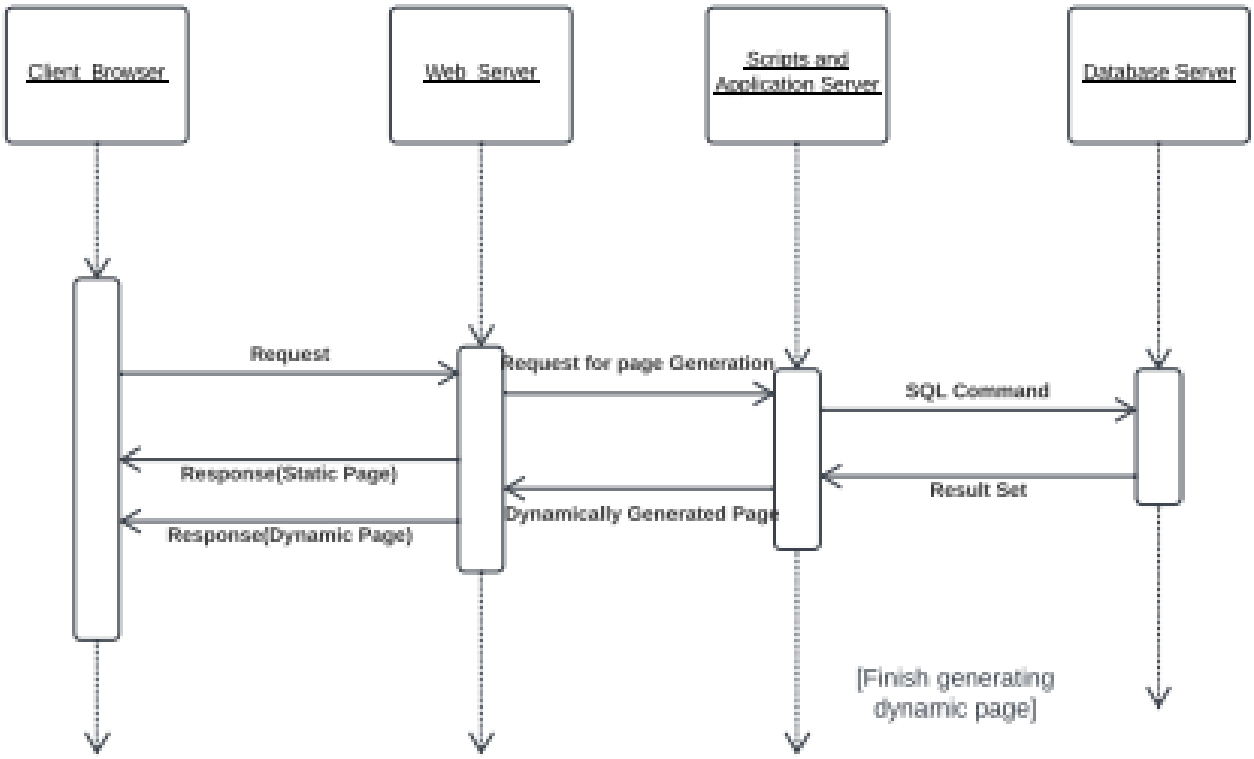

Figure 3: SEQUENCE DIAGRAM

4.COMMUNICATION DIAGRAM

AIM : To Draw A Communication Diagram For Web Scraping

Description : A communication diagram for web scraping is a visual representation of the interactions between the different components involved in the web scraping process. It shows the flow of messages exchanged between the user, the web scraper, and the target website(s). Here is a description of the main elements of a communication diagram for web scraping:

1. User: The user initiates the web scraping process by providing the URL(s) of the target website(s) to the web scraper.
2. Web Scraper: The web scraper receives the URL(s) from the user and sends an authentication request to verify the user's credentials.
3. User: The user responds to the authentication request by providing their credentials.
4. Web Scraper: The web scraper verifies the user's credentials and sends a request to the target website(s) to retrieve the HTML code.
5. Target Website: The target website receives the request from the web scraper and sends back the HTML code.
6. Web Scraper: The web scraper receives the HTML code and parses it to identify the relevant data.
7. Web Scraper: The web scraper extracts the relevant data from the HTML code and stores it in a database or a file.
8. User: The user may request the web scraper to extract additional data from the same or different websites by providing more URLs.
9. Web Scraper: The web scraper repeats the steps from 2 to 7 for each additional URL provided by the user.
10. Web Scraper: Once all the requested data has been extracted and stored, the web scraper sends a notification to the user indicating that the web scraping process is complete.

A communication diagram, also known as a collaboration diagram, is a type of interaction diagram that visualizes the flow of messages between objects or components in a system.

In the context of web scraping, a communication diagram could depict the interactions between various components involved in the process of scraping data from a website. This could include the web scraper tool itself, the web server hosting the website being scraped, and any intermediary systems such as proxies or middleware.

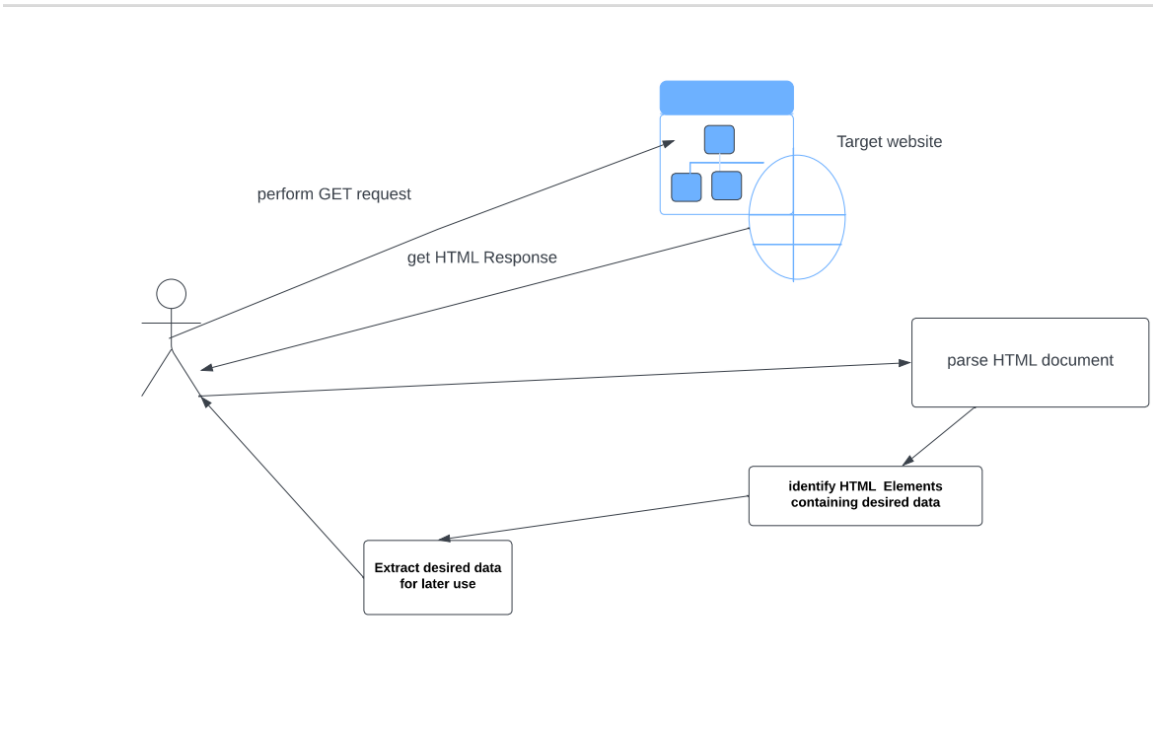

Figure 4: COMMUNICATION DIAGRAM

5.CLASS DIAGRAM

AIM : To Draw A Class Diagram For Web Scraping

Description : Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

- WebScraper: This class would be responsible for managing the overall process of web scraping. It could have methods for setting up connections to websites, sending requests, and parsing responses.
- Website: This class would represent a single website that the web scraper is interacting with. It could have properties such as a base URL and a list of pages to scrape.
- Page: This class would represent an individual page on a website. It could have properties such as a URL and a list of elements to extract.
- Element: This class would represent a specific piece of data on a page that the web scraper is interested in. It could have properties such as a selector (e.g. CSS selector or XPath expression) and a type (e.g. text, image, or link).
- Data: This class would represent the actual data that the web scraper retrieves from a website. It could have properties corresponding to the different types of elements that were extracted, such as a text property for text elements or a src property for image elements.
- DataStore: This class would be responsible for storing the data that the web scraper retrieves. It could have methods for adding new data and retrieving existing data.
- UserAgent: This class would represent the user agent that the web scraper is using to make requests to websites. It could have properties such as a browser type and version.

Overall, the class diagram for web scraping should reflect the various components of the scraping process, from connecting to the website and navigating to the relevant pages, to extracting and storing the data in a usable format.

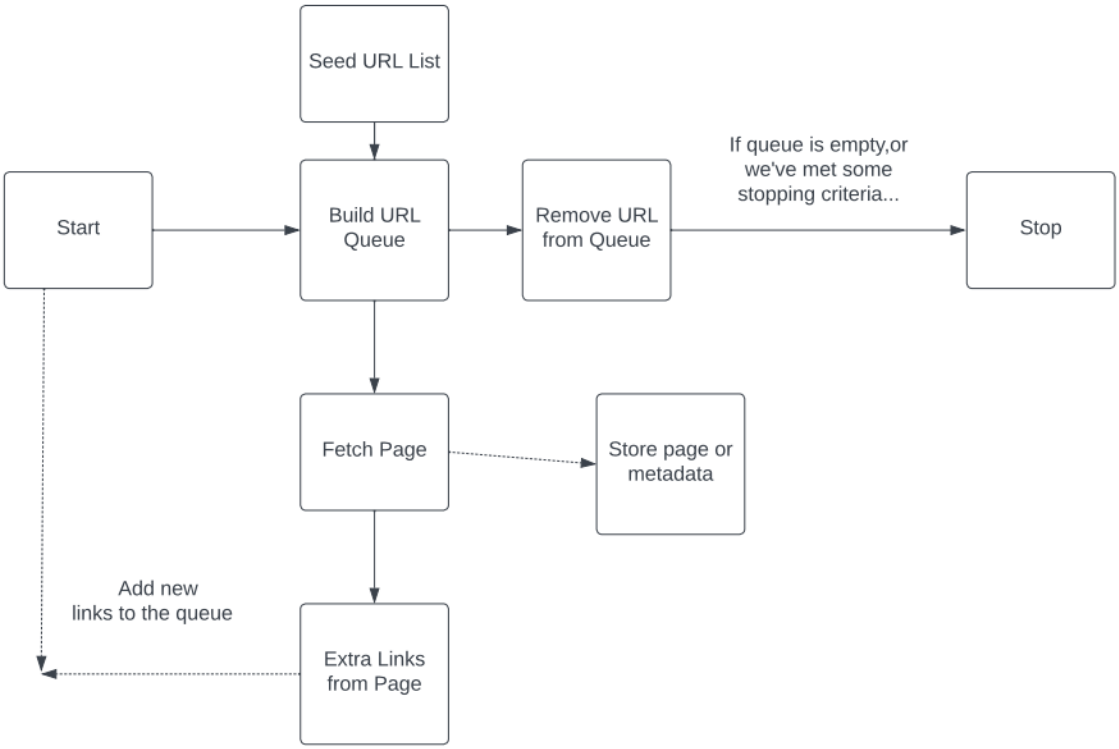

Figure 5: CLASS DIAGRAM

6.OBJECT DIAGRAM

AIM : To Draw A Object Diagram For Web Scraping

Description : An object diagram for web scraping is a snapshot of the system at a particular point in time, showing the objects and their relationships involved in the web scraping process. It represents a concrete instance of the class diagram and illustrates how the different classes interact with each other. Here is a description of the main elements of an object diagram for web scraping:

1. User: The user object represents the person who initiates the web scraping process by providing the URL(s) of the target website(s) to the web scraper.
2. User: The user object represents the person who initiates the web scraping process by providing the URL(s) of the target website(s) to the web scraper.
3. Authentication: The authentication object represents the process of verifying the user's credentials before allowing them to access the target website(s).
4. HTML Code: The HTML code object represents the markup language used to create web pages and is retrieved by the web scraper from the target website(s).
5. Data: The data object represents the information extracted by the web scraper from the HTML code of the target website(s).
6. Database/File: The database/file object represents the storage location where the extracted data is stored by the web scraper.

An object diagram for web scraping would illustrate the various objects involved in the process of extracting data from websites.

At the center of the diagram would be the web scraping tool or library being used, such as Beautiful Soup or Scrapy. This tool would be represented as an object with its own set of properties and methods.

Surrounding the web scraping tool would be objects representing the different elements on the web page being scraped. These could include HTML tags, CSS selectors, and XPath expressions, among others. Each of these objects would have its own properties, such as the element's class, ID, or value.

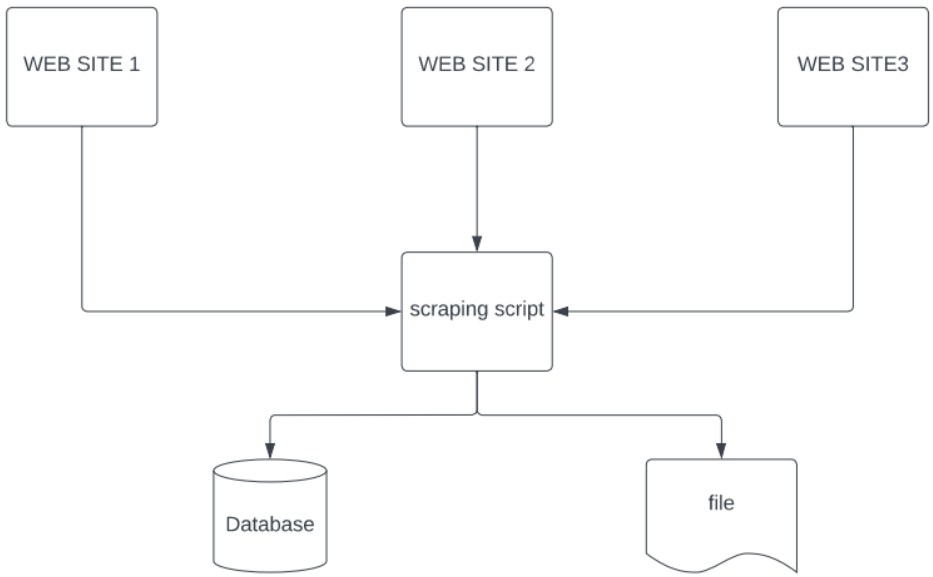

Figure 6: OBJECT DIAGRAM

7.COMPONENT DIAGRAM

AIM : To Draw A Component Diagram For Web Scraping

Description : A component diagram for web scraping is a structural view of the web scraping process that shows the components involved in the process and their relationships with each other. Here is a description of the main elements of a component diagram for web scraping:

1. User Interface Component: The user interface component represents the interface used by the user to initiate the web scraping process by providing the URL(s) of the target website(s) to the web scraper.
2. Web Scraper Component: The web scraper component represents the software application used to extract data from the target website(s).
3. Authentication Component: The authentication component represents the process of verifying the user's credentials before allowing them to access the target website(s).
4. HTML Code Component: The HTML code component represents the markup language used to create web pages and is retrieved by the web scraper from the target website(s).
5. Data Component: The data component represents the information extracted by the web scraper from the HTML code of the target website(s).
6. Database/File Component: The database/file component represents the storage location where the extracted data is stored by the web scraper.

A component diagram for web scraping would illustrate the different software components and their interactions required to perform web scraping operations.

At the center of the diagram would be the web scraping engine, which is responsible for carrying out the actual scraping of data from websites. It would be connected to several other components, such as the data processing module, which is responsible for cleaning and transforming the scraped data into a usable format.

Other components might include the user interface module, which provides a graphical interface for users to specify the websites to be scraped and the data to be extracted, and the scheduling module, which enables users to schedule the scraping of websites at specific times.

The component diagram might also include external components, such as databases or APIs, which are used to store the scraped data or to retrieve additional information to complement the scraped data.

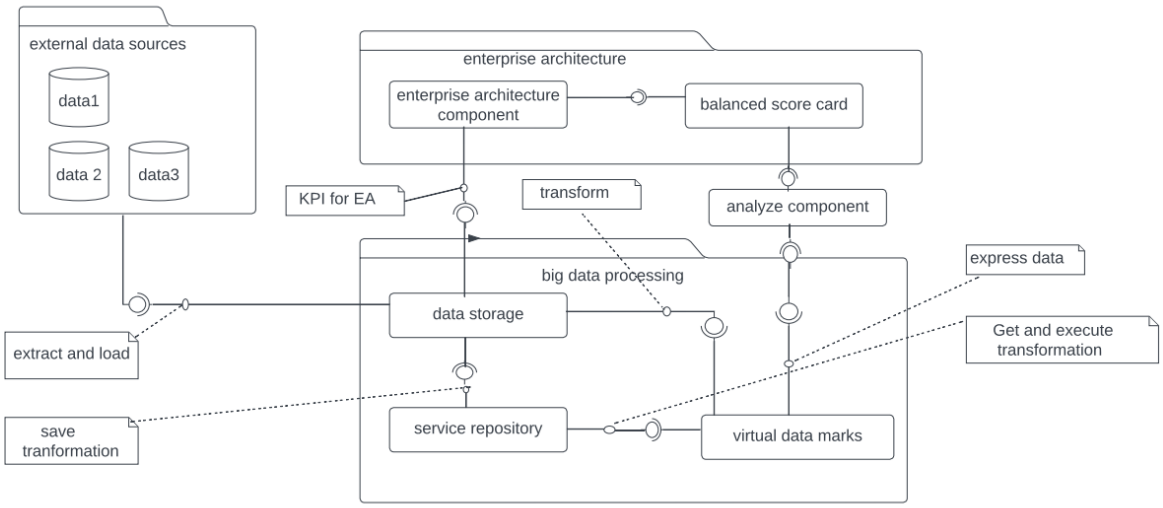

Figure 7: COMPONENT DIAGRAM

8.DEPLOYMENT DIAGRAM

AIM : To Draw A Deployment Diagram For Web Scraping

Description : A deployment diagram for web scraping is a structural view of the web scraping process that shows the physical deployment of the software and hardware components involved in the process. Here is a description of the main elements of a deployment diagram for web scraping:

1. User: The user represents the person who initiates the web scraping process by providing the URL(s) of the target website(s) to the web scraper.
2. Client Device: The client device represents the computer or mobile device used by the user to initiate the web scraping process.
3. Web Scraper: The web scraper represents the software application used to extract data from the target website(s).
4. Web Server: The web server represents the server where the target website(s) is hosted.
5. Database Server: The database server represents the server where the extracted data is stored by the web scraper.

A deployment diagram for web scraping would depict the physical architecture of the system used for scraping websites. It would show the various components of the system and how they are deployed, including servers, databases, and any other software or hardware necessary for the scraping process. The deployment diagram would also show the connections between these components, such as the communication between the web scraping client and the target web server, or the connection between the data storage system and the scheduler. Overall, the diagram would provide a clear visualization of the architecture and components involved in the web scraping process. The diagram would typically show the components involved in the web scraping process, such as the web scraper application, web servers, databases, and other relevant resources. The components may be distributed across multiple servers or systems, depending on the complexity of the web scraping operation.

The deployment diagram would also show the communication channels between components, such as APIs, web services, or messaging queues, and the protocols used for data exchange, such as HTTP, TCP/IP, or other relevant standards.

The diagram may also show how the web scraper application is deployed and managed, including the use of containers, virtual machines, or cloud services, and the configuration of the system for scalability, fault tolerance, and performance optimization.

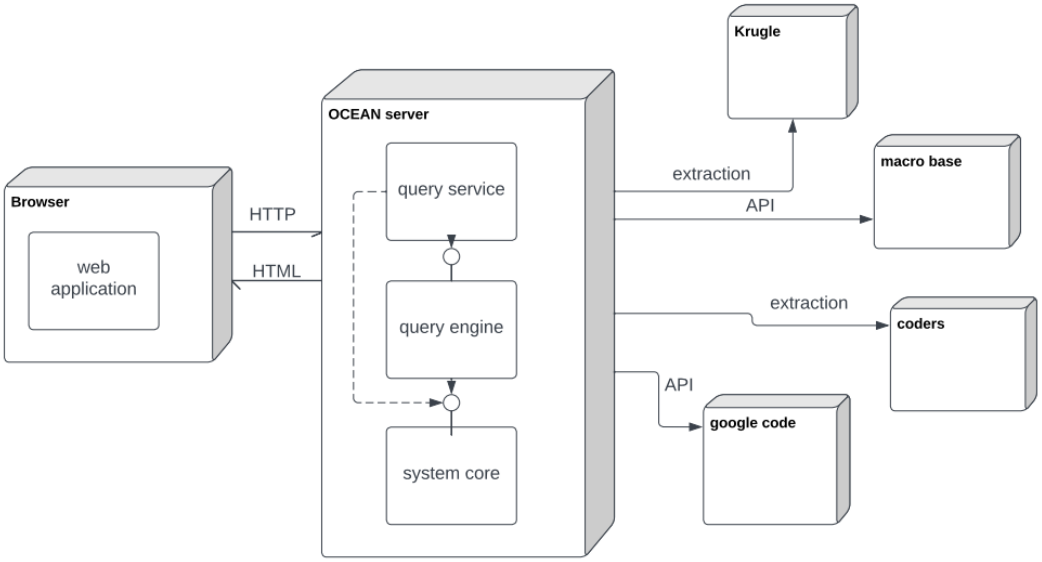

Figure 8: DEPLOYMENT DIAGRAM

9.STATE MACHINE DIAGRAM

AIM : To Draw A State Machine Diagram For Web Scraping

Description : A state machine diagram for web scraping is a behavioral view of the web scraping process that shows the different states and transitions of the web scraper during the scraping process. Here is a description of the main elements of a state machine diagram for web scraping:

1. Idle State: The idle state represents the initial state of the web scraper when it is not scraping any website.
2. Login State: The login state represents the state of the web scraper when it is authenticating the user's credentials to access the target website(s).
3. Scraping State: The scraping state represents the state of the web scraper when it is actively scraping the target website(s) to extract data.
4. Parsing State: The parsing state represents the state of the web scraper when it is processing the HTML code of the target website(s) to extract the relevant data.
5. Saving State: The saving state represents the state of the web scraper when it is storing the extracted data in a database or a file.
6. Error State: The error state represents the state of the web scraper when an error occurs during the scraping process.

The initial state is usually the "Start" state, where the program begins executing. From there, the program typically transitions to the "Connect" state, where it establishes a connection to the website to be scraped.

The next state is often the "Retrieve" state, where the program retrieves the HTML code from the website. This may involve parsing the HTML code to extract the relevant data.

Once the data has been extracted, the program may transition to the "Transform" state, where it transforms the data into a more usable format, such as a CSV or JSON file.

Finally, the program may transition to the "Store" state, where it stores the transformed data in a database or file for later use.

Throughout the process, the program may encounter error states, such as "Connection Failed" or "Parsing Error," which require specific actions to be taken to resolve the issue and continue with the scraping process.

Overall, the state machine diagram for web scraping provides a visual representation of the various states and transitions involved in the process, making it easier to understand and debug the program.

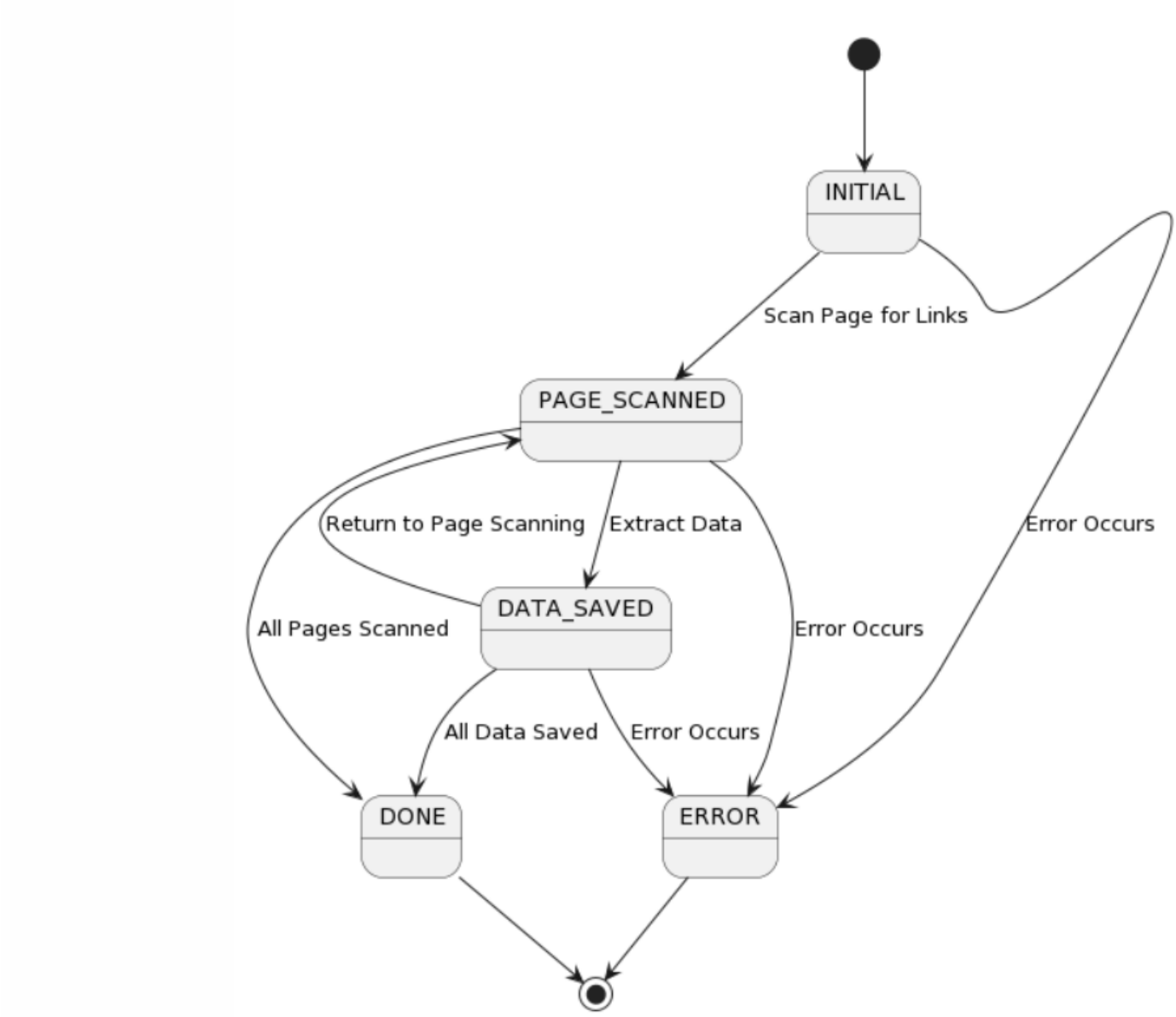

Figure 9: STATE MACHINE DIAGRAM

COCOMO MODEL FOR WEB SCRAPING

AIM: Web Scraping using COCOMO estimation effort.

DESCRIPTION: Boehm proposed COCOMO (Constructive Cost Estima- tion Model) in 1981.COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

2.1 The necessary steps in this model are:

1. Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code(KDLOC).
2. Determine a set of 15 multiplying factors from various attributes of the project.
3. Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort in person-months the equation used is of the type is shown below

   Ei=a*(KDLOC)b The value of the constant a and b are depends on the project type.

In COCOMO, projects are categorized into three types:

1. Organic
2. Semidetached
3. Embedded

1. Organic: A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. Examples of this type of projects are simple business systems, simple inventory management systems, and data processing systems.

2. Semidetached: A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. Examples of this type of projects are simple business systems, simple inventory management systems, and data processing systems.

3. Embedded: A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hard- ware, or if the stringent regulations on the operational method exist.

For Example: ATM, Air Traffic control.

For three product categories, Bohem provides a different set of expression to predict effort (in a unit of person month)and development time from the size of estimation in KLOC(Kilo Line of code) efforts estimation takes into account the productivity loss due to holidays, weekly off, coffee breaks, etc.

According to Boehm, software cost estimation should be done through three stages:

1. Basic Model
2. Intermediate MOdel
3. Detalied Model

2.1.1 Basic COCOMO Model:

The basic COCOMO model provide an accurate size of the project parameters. The following expressions give the basic COCOMO estimation model:

- Effort=a1*(KLOC) a2 PM
- Tdev=b1*(efforts)b2 Months

Where,KLOC is the estimated size of the software product indicate in Kilo Lines of Code,a1,a2,b1,b2 are constants for each group of software prod- ucts,Tdev is the estimated time to develop the software, expressed in months,Effort is the total effort required to develop the software product, expressed in per- son months (PMs). Estimation of development

effort For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

- Organic: Effort = 2.4(KLOC) 1.05 PM
- Semi-detached: Effort = 3.0(KLOC) 1.12 PM
- Embedded: Effort = 3.6(KLOC) 1.20 PM

2.1.2 Intermediate Model:

The basic Cocomo model assumes that the effort is only a function of the number of lines of code and some constants evaluated according to the different software systems. However, in reality, no system's effort and schedule can be solely calculated on the basis of Lines of Code. For that, various other factors such as reliability, experience, Capability. These factors are known as Cost Drivers and the Intermediate Model utilizes 15 such drivers for cost estimation.

beginfigure[htp]

Project	a_i	b_i	c_i	d_i
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

2.1.3 Detailed COCOMO Model:

Detailed COCOMO incorporates all qualities of the standard version with an assessment of the cost driver?s effect on each method of the software engi- neering process. The detailed model uses various effort multipliers for each cost driver property. In detailed cocomo, the whole software is differentiated into multiple modules, and then we apply COCOMO in various modules to estimate effort and then sum the effort.

The Six phases of detailed COCOMO are:

1. Planning and requirements
2. System structure
3. Complete structure
4. Module code and test
5. Integration and test
6. Cost Constructive model

The effort is determined as a function of program estimate, and a set of cost drivers are given according to every phase of the software lifecycle.

YOUR BASIC COCOMO RESULTS!!

MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person/months)	DURATION, (in months)	STAFFING, (recommended)
embedded	3.6	1.2	2.5	0.32	3	13.453894147847587	5.743430507902744	2.342484013576126

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm,81). The final estimates are determined in the following manner:

effort =a*KLOC^{b}, in person/months, with KLOC = lines of code, (in the thousands), and:

duration =c*effort^{d}, finally:

staffing =effort/duration

For further reading, see Boehm, "Software Engineering Economics",(81)

WARNING: If you see "NaN" in any field above, you have entered an **INVALID** value for KLOC!! Hit the "BACK" button on your browser, hit the "RESET" button, and enter a **DECIMAL NUMBER** in the KLOC input text box!

Thank you, and happy software engineering!

ENTITY RELATIONSHIP DIAGRAM ON WEB SCRAPING

Aim : To Draw Entity Relationship Diagram For Web scraping

Description : Web scraping is the process of automatically extracting data from websites. Here is an entity relationship diagram that illustrates the main entities involved in a typical web scraping process:

Explanation :

- A website is a top-level entity that represents a website that contains data to be scraped. It contains information about the website, such as its name, base URL, and scraper configuration.
- A web page is a second-level entity that represents a specific page on the website. It contains information about the URL and the HTML content of the page.
- An HTML element is a third-level entity that represents a specific element on the web page that needs to be scraped. It contains information about the tag name and CSS selector used to locate the element on the page.
- A data attribute is a fourth-level entity that represents a specific attribute of an HTML element that contains the data to be scraped. It contains information about the name and value of the attribute.

  Each web page belongs to a website, each HTML element belongs to a web page, and each data attribute belongs to an HTML element. Therefore, there are foreign key relationships between the entities, as indicated by the (FK) notation. This allows for easy retrieval of data from a specific website, web page, HTML element, or data attribute.

Note: This is just one example of an entity relationship diagram for web scraping, and there are many variations depending on the specific requirements of the project.

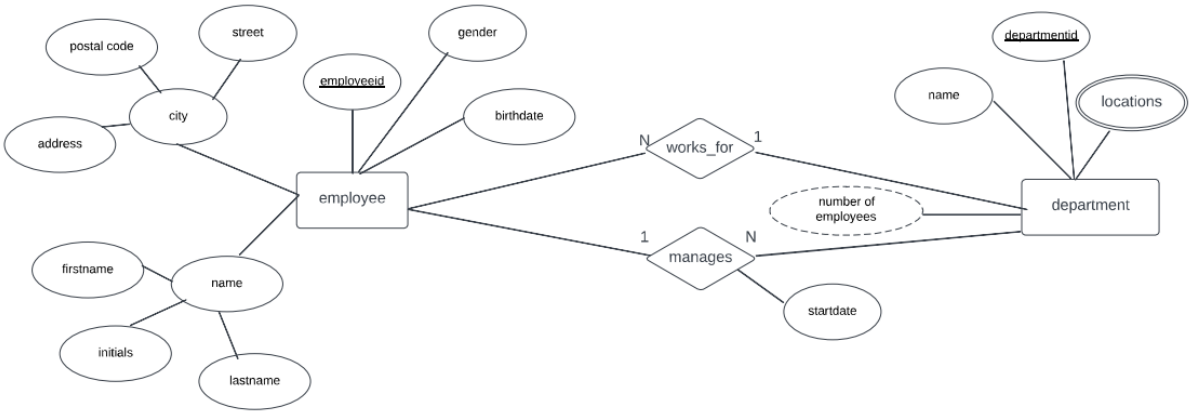

Figure 10: ENTITY RELATIONSHIP DIAGRAM

DATA FLOW DIAGRAM ON WEB SCRAPING

Aim : To Draw A Data Flow Daigram For Web Scraping

Description : A data flow diagram (DFD) for web scraping typically illustrates the flow of data and actions involved in the process of extracting information from a website.

At a high level, the DFD may include the following components:

- Data Sources: These are the websites or web pages from which data is to be extracted.
- Web Scraper: This is the program or script that is used to extract data from the websites. It sends HTTP requests to the websites and receives HTML responses that are parsed to extract relevant information.
- Data Storage: This component includes the database or data storage system where the extracted data is stored for future use.
- Data Consumers: These are the users or applications that will use the extracted data. Examples of data consumers include data analysts, machine learning models, and other software systems.

The DFD may also include additional components such as data cleaning and transformation processes, as well as error handling and logging mechanisms to ensure the integrity and reliability of the extracted data.

Overall, the DFD for web scraping is a visual representation of the different components involved in the process of extracting data from websites, and how they interact with each other to achieve the desired outcome.

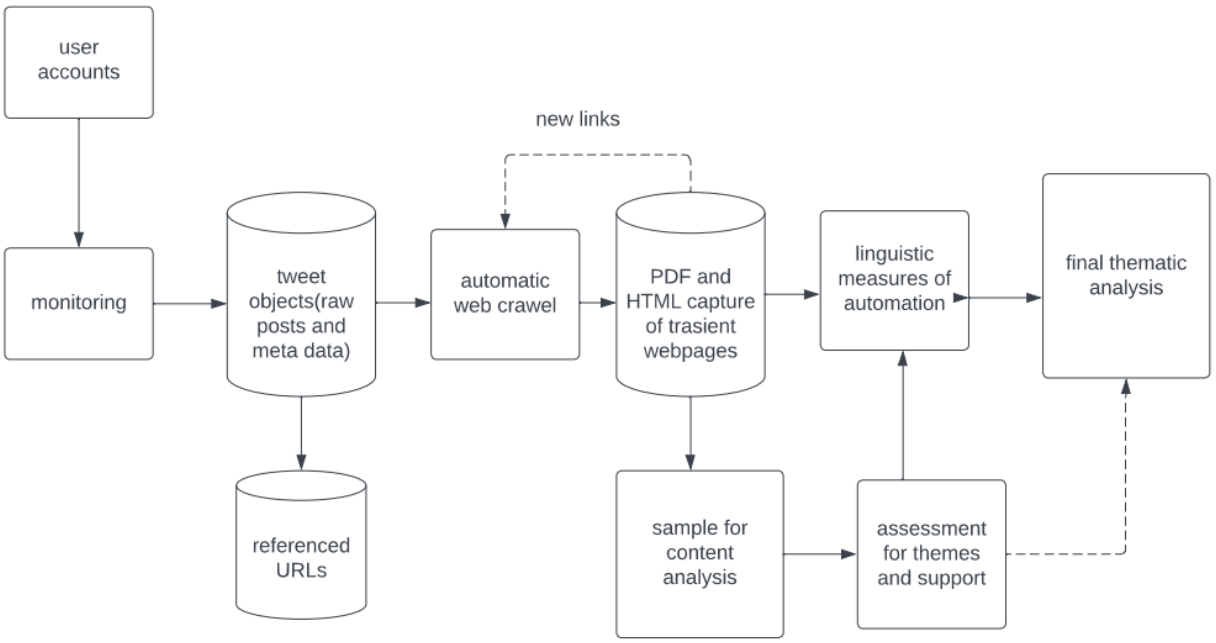

Figure 11: DATA FLOW DIAGRAM

CONTROL FLOW DIAGRAM ON WEB SCRAPING

Aim : To Draw A Control Flow Diagram For Web Scraping

Description : A control flow diagram for web scraping typically shows the steps involved in the process of scraping data from a website. The diagram typically begins with an initial step of identifying the website to be scraped. This is followed by a step to identify the specific data to be extracted from the website.

Once the data to be extracted has been identified, the next step is to set up a web scraper or crawler to navigate the website and extract the data. The scraper will typically follow a specific set of rules to ensure that the data is extracted correctly and that the process is efficient.

As the scraper navigates the website, it may encounter various obstacles such as CAPTCHA tests or other security measures designed to prevent web scraping. The control flow diagram should include steps to handle these obstacles, such as temporarily pausing the scraper until the obstacle can be overcome.

Once the data has been successfully extracted, the next step is to store the data in a suitable format, such as a database or spreadsheet. The control flow diagram should include steps to handle any errors that may occur during the storage process, such as data formatting errors or database connection failures.

Finally, the diagram should include steps for cleaning and processing the data to ensure that it is accurate and useful. This may involve removing duplicates, filling in missing data, or performing other data transformations.

Overall, a control flow diagram for web scraping should provide a clear and systematic overview of the entire process, from identifying the website to be scraped to cleaning and processing the extracted data.

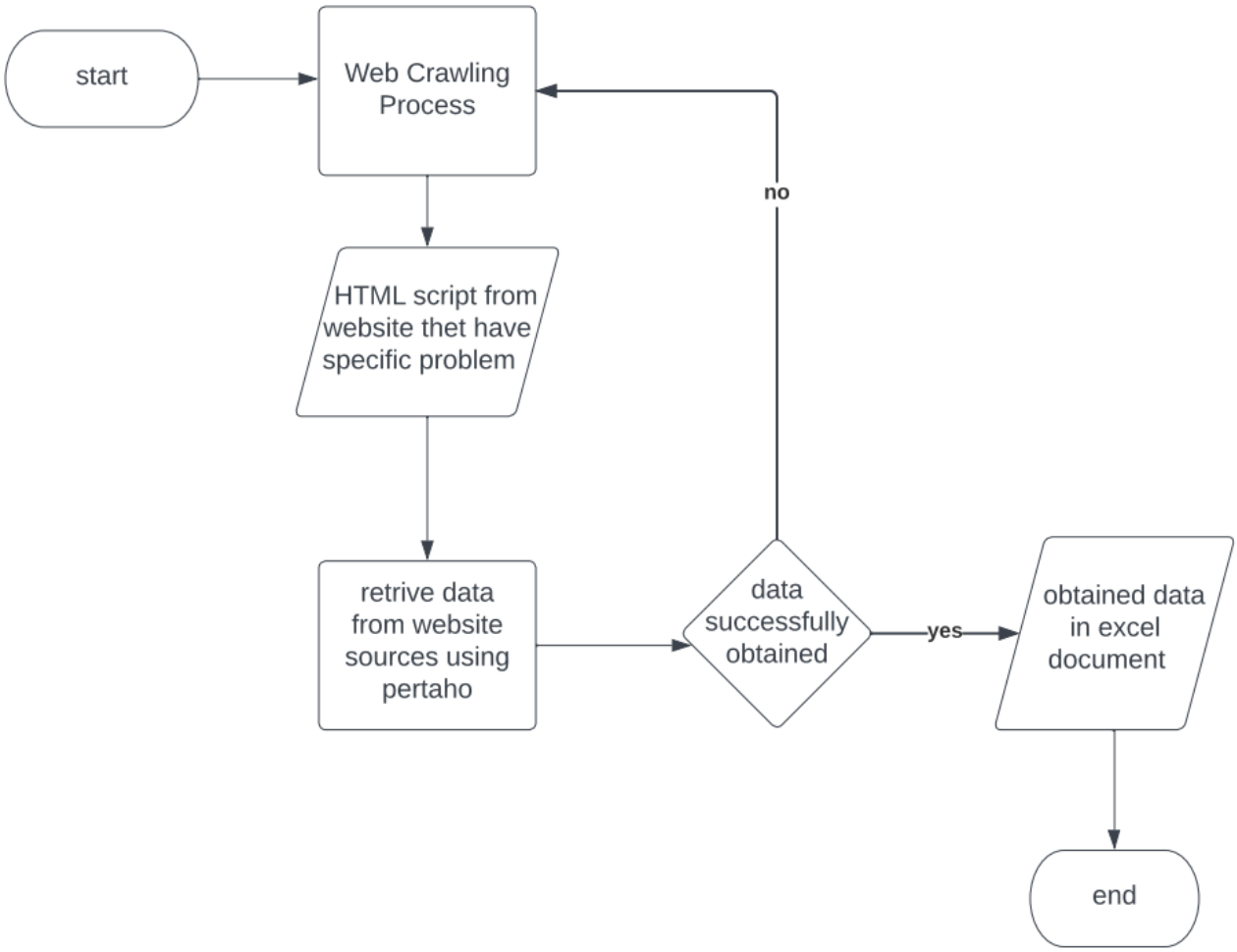

Figure 12: CONTROL FLOW DIAGRAM

STRUCTURED CHARTS FOR WEB SCRAPPING

AIM : To draw Structured charts for web scraping.

DESCRIPTION : Structured charts are a helpful tool for organizing and documenting the web scraping process. They can help you plan out the steps needed to extract data from a website and ensure that you don't miss any important details.

Here is an example of a structured chart for web scraping:

1. Identify the website to scrape
2. Identify the data to be extracted
3. Determine the data extraction method
4. Create a scraping script
5. Test the scraping script
6. Refine the scraping script
7. Run the scraping script
8. Export the extracted data
9. Clean and format the data

This structured chart can be customized to fit your specific needs and can be used as a guideline for your web scraping project. It's important to note that web scraping can raise ethical and legal concerns, so be sure to check the website's terms of use and respect their policies.

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. The web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol or through a web browser.

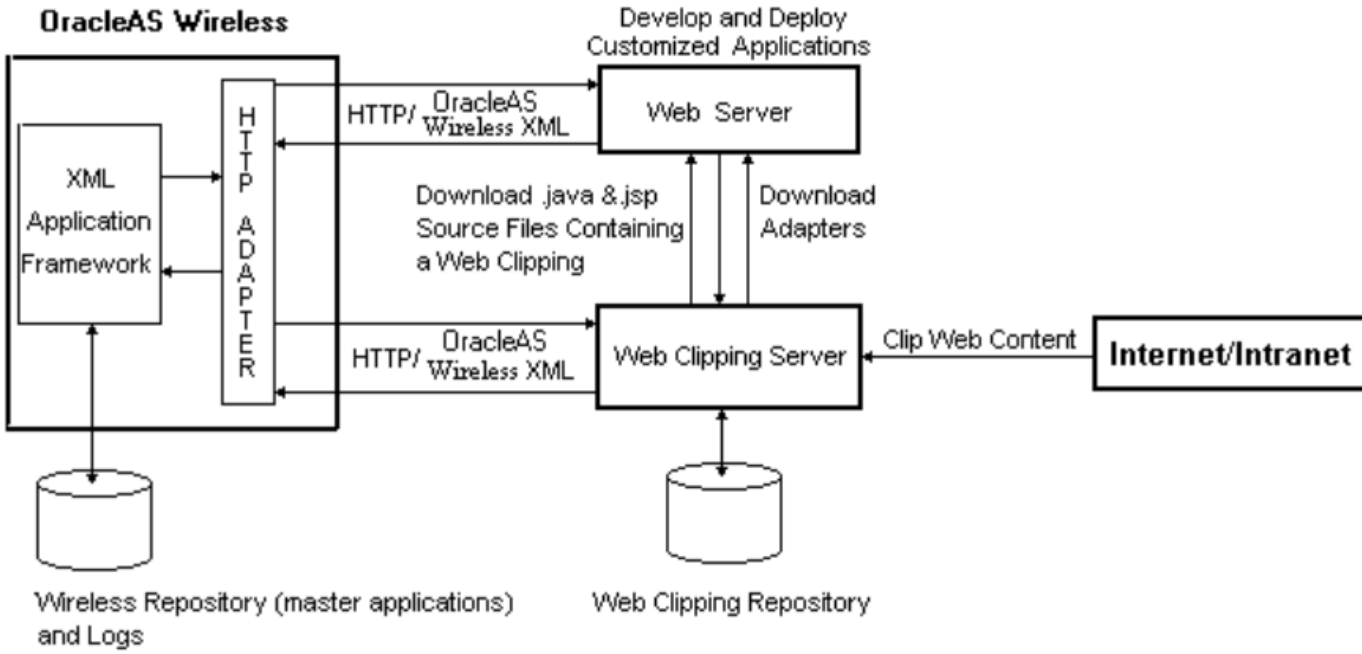

Figure 13: STRUCTURED CHART DIAGRAM