``````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````

***Chapter 1***

# INTRODUCTION

**s**

## 1.1 - PROJECT OVERVIEW

The project, **“Automated Flowchart Visualization for Understanding Complex Relations,”** is designed to address the challenges associated with navigating and analyzing large amounts of unstructured data from multiple online sources. The system will automatically extract data using web scraping techniques, process the data using Natural Language Processing (NLP) methods to identify key entities, and visualize these relationships in the form of dynamic flowcharts.

The platform leverages tools like SERP for data collection, while Python-based NLP libraries such as NLTK will be used for entity extraction and relationship analysis. The flowcharts are generated using Graphviz, offering users interactive experience with clickable nodes that lead to external resources for further exploration. The web interface will be built using React, ensuring a responsive and dynamic user experience. The platform will also be hosted on cloud services like to ensure scalability and accessibility.

This project aims to make data more navigable and understandable, providing a clear visual representation of complex relationships and serving as an educational tool for research and learning.

## - NEED OF PROJECT

In a world inundated with information, finding meaningful connections between various entities such as people, places, and events has become a daunting task. Traditional search engines and methods of data retrieval are insufficient when it comes to representing these complex relationships in a visual and user-friendly format. Users often face challenges in organizing and interpreting unstructured data from multiple sources, which can lead to incomplete or inaccurate conclusions.

There is a growing need for a system that not only extracts relevant information but also presents it in a way that allows users to see the connections between different entities at a glance. A flowchart-based approach, which visualizes these relationships and provides links to external resources, is an effective solution. This project addresses this need by automating the data collection, processing, and visualization processes, making it easier for users to explore large datasets and uncover valuable insights without the need for manual navigation.

## - LITERATURE SURVEY

The fields of web scraping and data extraction have seen significant advancements in recent years, with tools like **Serp** playing a pivotal role in automating the retrieval of data from various online sources. These tools, when combined with **Natural Language Processing (NLP)** techniques like Named Entity Recognition (NER), enable the extraction of key entities and relationships from unstructured text. According to research, NER is a crucial technique used to identify and classify entities such as people, locations, and organizations, making it highly relevant to projects involving entity relationship extraction.

In terms of data visualization, flowchart generation has become an increasingly popular method for simplifying the representation of complex data. Tools like **Graphviz** are widely used for programmatically creating flowcharts, offering dynamic and interactive visualizations on web platforms. These tools are particularly beneficial in representing the relationships between different entities, making them ideal for this project’s requirements. Research shows that interactive user interfaces with embedded hyperlinks and dynamic content significantly enhance user engagement and understanding, particularly in educational and research contexts.

The literature also highlights the importance of integrating web scraping, NLP, and visualization techniques to create a cohesive system for presenting interconnected data. Combining these technologies allows for an automated, scalable solution to represent complex relationships, aligning with the objectives of this project.

***Chapter 2***

# PROBLEM DEFINITION AND SCOPE

## - PROBLEM STATEMENT

In the modern digital age, information related to people, locations, and events is widely available across numerous online platforms such as Wikipedia, news websites, and public archives. However, the sheer volume and unstructured nature of this data make it difficult to analyze and understand the relationships between these entities. Users often need to manually sift through vast amounts of information, making it time- consuming and prone to error.

Traditional search methods and information retrieval systems are inadequate for visualizing the complex web of connections that exist between different entities, leading to a fragmented understanding of the data. This lack of cohesive visualization results in inefficiencies in research and analysis, making it challenging for users to uncover meaningful insights or explore the contextual background of entities in an intuitive way. Therefore, there is a need for a platform that can automate the extraction, processing, and visualization of data, allowing users to easily navigate through complex relationships using an interactive flowchart-based interface.

## – SCOPE

The scope of this project includes the development of a fully functional web-based platform that automates data extraction, processing, and visualization.

The system will be capable of:

* + - **Web Scraping**: Collecting data from multiple sources such as Wikipedia and news websites.
    - **Natural Language Processing (NLP)**: Extracting and classifying key entities such as people, places, and events using Named Entity Recognition (NER) techniques.
    - **Flowchart Generation**: Creating dynamic, interactive flowcharts using tools like Graphviz.
    - **Interactive Web Interface**: Building a user-friendly interface using React that allows users to navigate complex relationships visually.

The platform is targeted towards educational and research institutions, as well as professionals in the media industry who require a clear representation of data relationships for analysis and decision-making. The project will focus on maintaining scalability, real-time data processing, and engaging user experience.

## - AREA OF PROJECT

This project falls within the interdisciplinary domains of:

* + - **Data Science**: Using data extraction, NLP, and visualization techniques to analyze and represent large datasets.
    - **Web Development**: Building an interactive front-end interface that visualizes complex data in an intuitive format.
    - **Information Retrieval**: Collecting, processing, and presenting data from various online sources using web scraping techniques.
    - **Data Visualization**: Representing complex relationships using tools like Graphviz, which are popular in creating entity-relationship diagrams.

## - GOALS AND OBJECTIVES

### Goals:

The primary goal of the project is to create an automated system that transforms

complex, unstructured data into interactive flowcharts, allowing users to explore and understand the relationships between various entities more intuitively. The project aims to streamline data exploration and make information more accessible through visual representation.

### Objectives:

1. **Automate Data Collection**: Implement web scraping to collect data from multiple online sources such as Wikipedia, news sites, and public archives.
2. **Extract & Analyze Relationships**: Use NLP techniques to extract key entities and classify them into categories such as people, locations, and events.
3. **Generate Interactive Visualizations**: Create dynamic flowcharts using Graphviz to visually represent the connections between entities.
4. **Develop a Web-Based Interface**: Build a responsive and user-friendly interface for exploring the generated flowcharts, using React or Next.js.

***Chapter 3***

# SOFTWARE REQUIREMENT SPECIFICATION

## - SOFTWARE REQUIREMENT

To develop the **"Automated Flowchart Visualization for Understanding Complex Relations"** platform, the following software tools and technologies are required:

* + - **Web Scraping Libraries**:
      * **SERP**: A powerful web crawling framework.
      * **Cheerio**: A JavaScript library for parsing HTML and extracting data.
    - **Natural Language Processing (NLP) Libraries**:
      * **NLTK**: Libraries for performing Named Entity Recognition (NER) and analyzing relationships between entities.
    - **Programming Language**: Python for NLP processing, and JavaScript for web scraping and front-end development.
    - **Flowchart Generation Tools**:
      * **Graphviz:** Open-source graph visualization software used to render graphs / flowcharts.
    - **Web Development Framework**:
      * **React.js**: JavaScript frameworks for building a dynamic, responsive web platform.
    - **Database**:
      * **MongoDB**: For storing and managing scraped and processed data.
    - **Version Control**:
      * **Git**: For managing source code, tracking changes, and collaborating with team members.
    - **API Testing Tools**:
      * **Postman**: For testing and validating the platform's backend APIs.

## - HARDWARE REQUIREMENT

For smooth development and testing of the platform, the following hardware resources are recommended:

* + - **Development System**:
      * **Processor**: Intel Core i5 (or equivalent) for basic operations; Intel Core i7 or higher for faster processing.
      * **RAM**: 8 GB minimum; 16 GB recommended for better performance during data processing and development.
      * **Storage**: 256 GB SSD (minimum); 512 GB or higher recommended for efficient data handling and faster development environment setup.
    - **Network Requirements**:
      * **Stable Internet Connection**: A minimum speed of 10 Mbps for downloading libraries, scraping data from online sources, and cloud-based tasks.

## - FUNCTIONAL REQUIREMENT

The functional requirements define the key features and behaviors of the system:

1. **Data Collection**:
   * The system should scrape relevant data from multiple online sources, such as Wikipedia and news archives, using web scraping libraries like SERP and Cheerio.
   * The platform should ensure accurate data collection by handling various web page structures and formats.
2. **Entity Extraction**:
   * The platform should use NLP techniques (NLTK) to process the scraped data and extract key entities, such as people, locations, and events.
   * The system should classify these entities using Named Entity Recognition (NER) and map relationships between them.
3. **Flowchart Generation**:
   * The system should generate dynamic, interactive flowcharts representing the relationships between entities using tools like Graphviz.
   * The flowcharts should include clickable nodes that link to additional resources for users to explore further.
4. **Interactive User Interface**:
   * The web platform should provide a responsive and intuitive interface, built using React or Next.js, allowing users to visualize and interact with the flowcharts.
   * Users should be able to navigate through relationships, zoom in/out of the flowcharts, and click on nodes to access external resources.
5. **Data Storage**:
   * The system should store the scraped and processed data in a local database for efficient retrieval and use in flowchart generation.
   * The database should manage and store entity relationships, scraped content, and user interactions.
6. **Testing and Validation**:
   * The platform should have robust testing mechanisms using Postman or similar tools to validate backend API functionality and ensure data accuracy.
   * Continuous integration and testing should be implemented to ensure smooth platform updates and bug fixes.

***Chapter 4***

# PROJECT PLAN

## - PROJECT SCHEDULE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Tasks** | **Duration** | **Start Day** | **End Day** |
| 1. Requirement Gathering | * Define project objectives and scope. * Collect requirements from stakeholders. * Identify key system functionalities and resources. * Finalize the requirements document. | 2 weeks | Day 1 | Day 14 |
| 2. Design Phase | * Design system architecture. * Create database schema * Design flowchart generation model. * Design user interface mockups * Finalize system designs. | 4 weeks | Day 15 | Day 42 |
| 3. Detailed Designing Phase | * Develop web scraping module. * Research NLP engine for entity extraction. * Understand entity-relationship processing. * Develop flowchart generation module. * Build front-end wireframes. * Perform internal testing of all modules. | 6 weeks | Day 43 | Day 84 |

* 1. **- PROJECT COST ESTIMATION**

The project’s cost estimation includes software, hardware, and hosting costs. As the project involves using open-source tools and minimal hardware, the estimated cost is primarily related to cloud hosting and development tools.

|  |  |  |
| --- | --- | --- |
| **Category** | **Details** | **Estimated Cost (INR)** |
| Hardware | Development laptop (if required) | - |
| Software | Open-source tools (SERP, Cheerio, etc.) | - |
| Testing Tools | Postman, browser testing | - |
| Flowchart generator | Graphviz for converting text into graph | 1000 INR |
| **Total Estimated Cost** |  | **1000 INR** |

***Chapter 5***

# SOFTWARE DESIGN

## - DATA FLOW DIAGRAM

* + 1. **- LEVEL 0 DATA FLOW DIAGRAM**

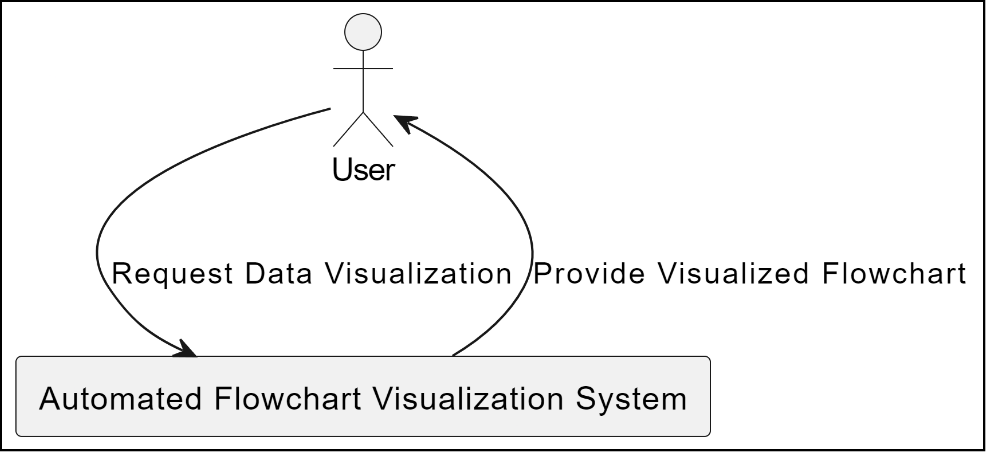
****

Fig.5.1.1: Level 0 data flow diagram

* + 1. **- LEVEL 1 DATA FLOW DIAGRAM**

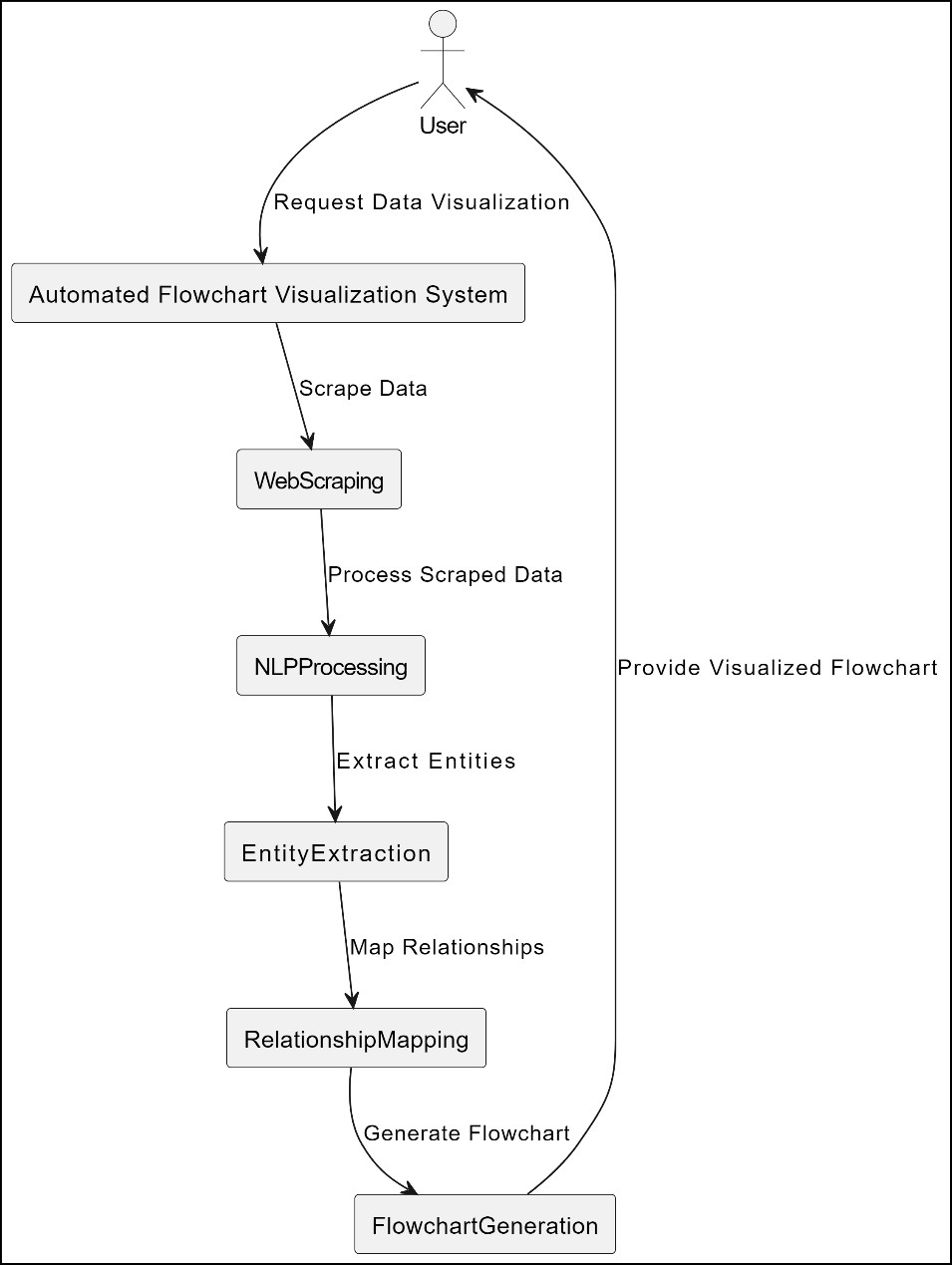
****

Fig.5.1.2: Level 1 data flow diagram

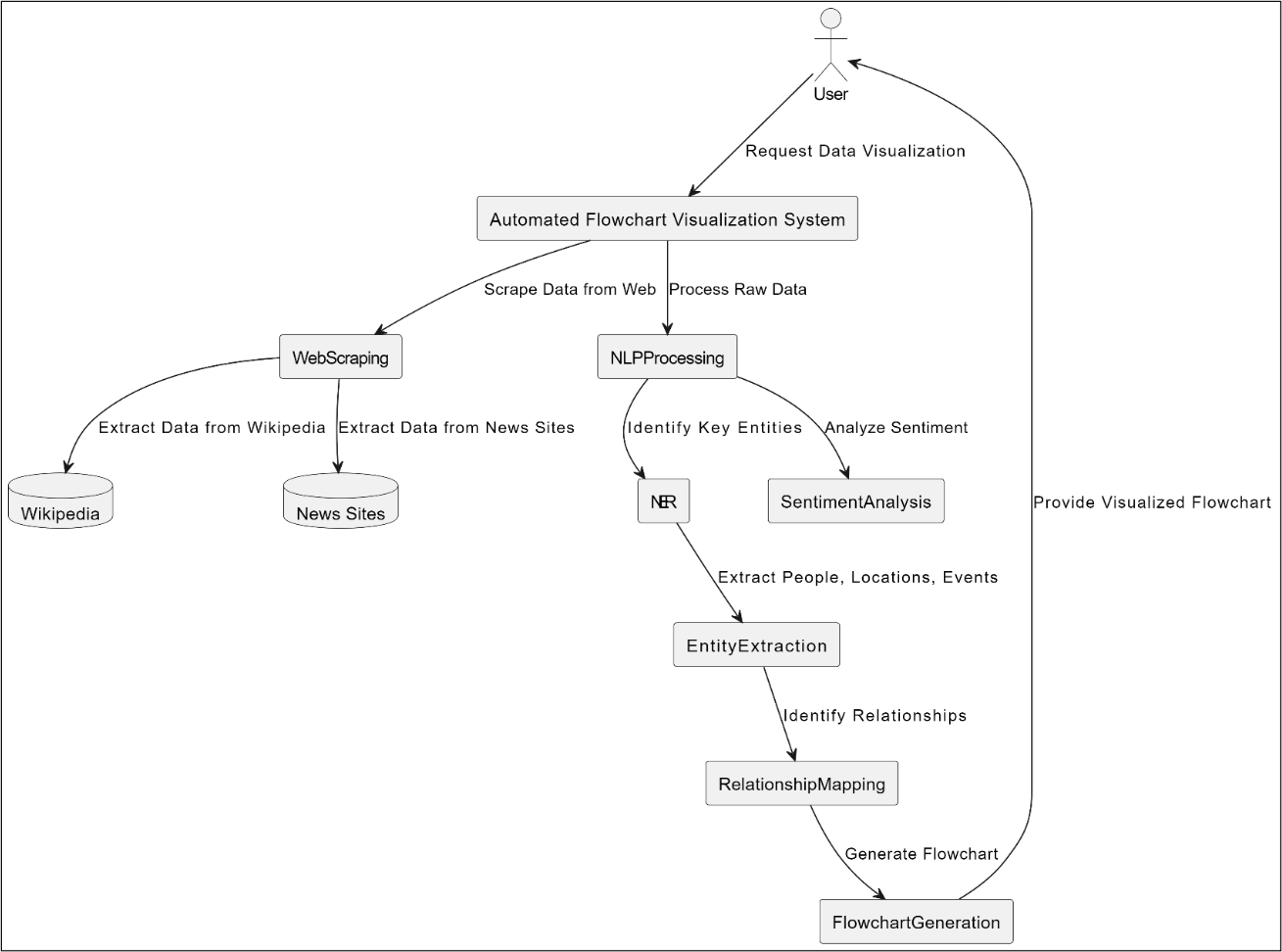
* + 1. **– LEVEL 2 DATA FLOW DIAGRAM**

Fig.5.1.3: Level 2 data flow diagram

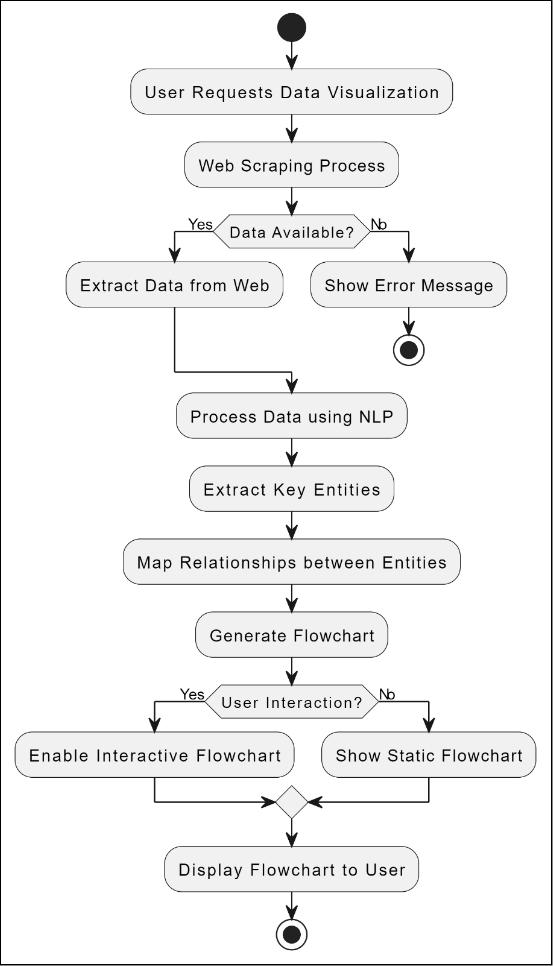
* 1. **- FLOW CHART**

Fig.5.2: Flow chart

* 1. **- SYSTEM ARCHITECTURE**

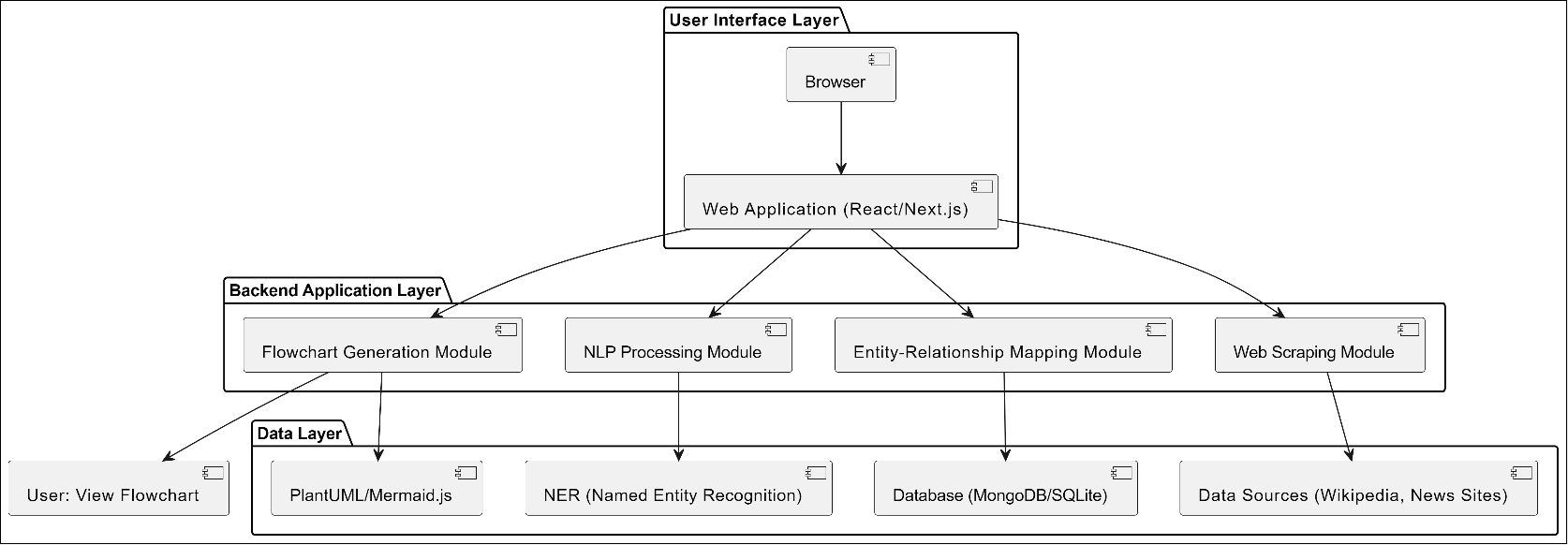
****

Fig.5.3: System architecture

* 1. **- UML DIAGRAMS**
     1. **- ER DIAGRAM**

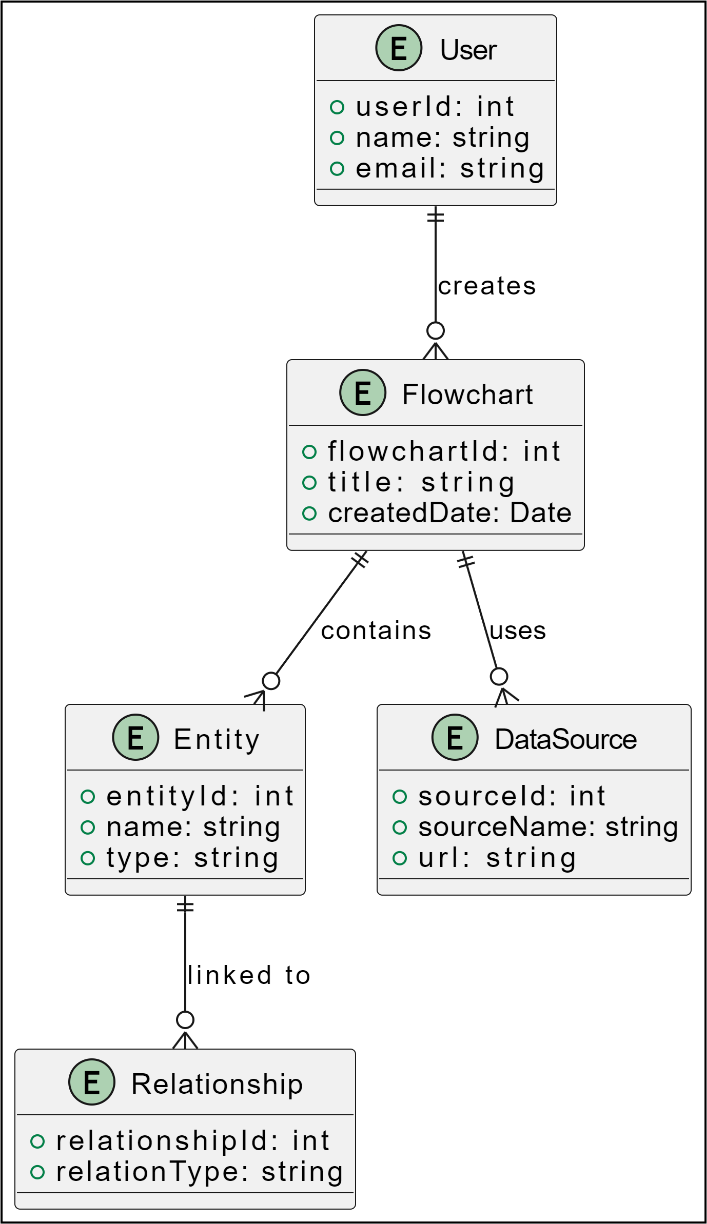
****

Fig.5.4.1: ER diagram

* + 1. **- USE CASE DIAGRAM**

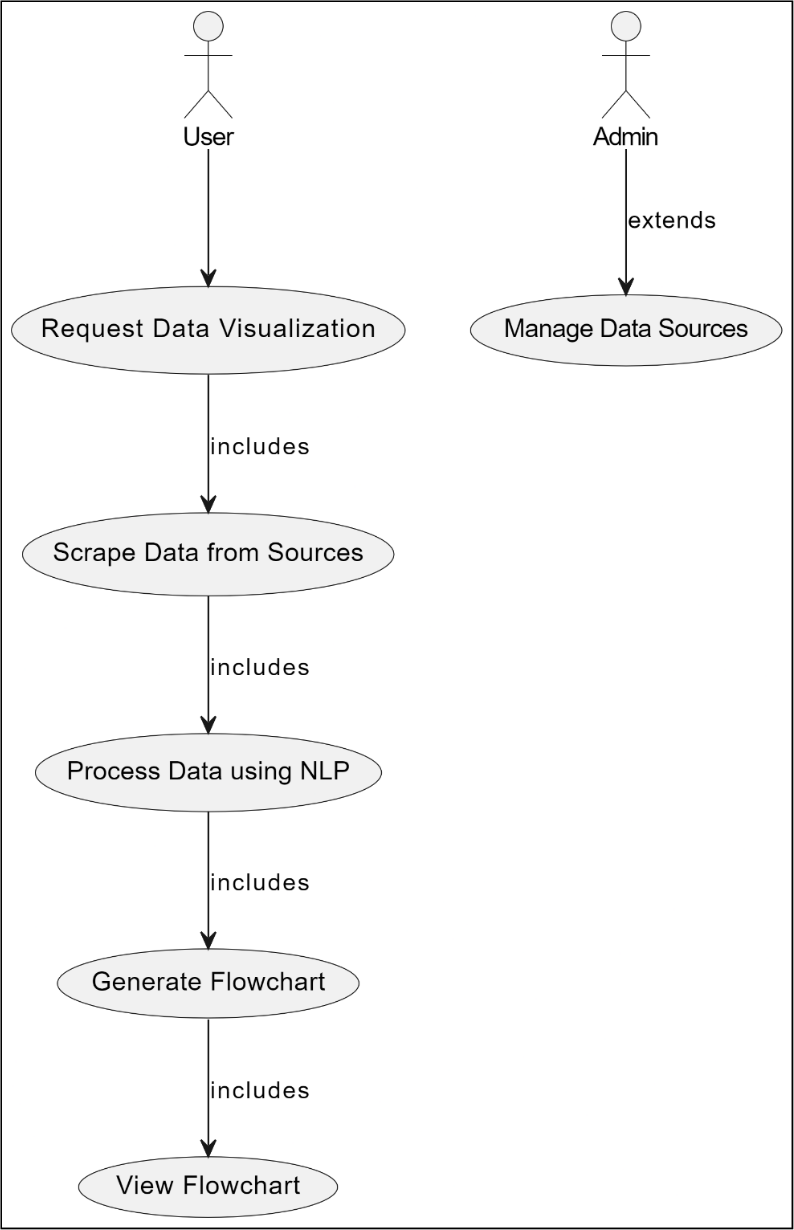
****

Fig.5.4.2: Use case diagram

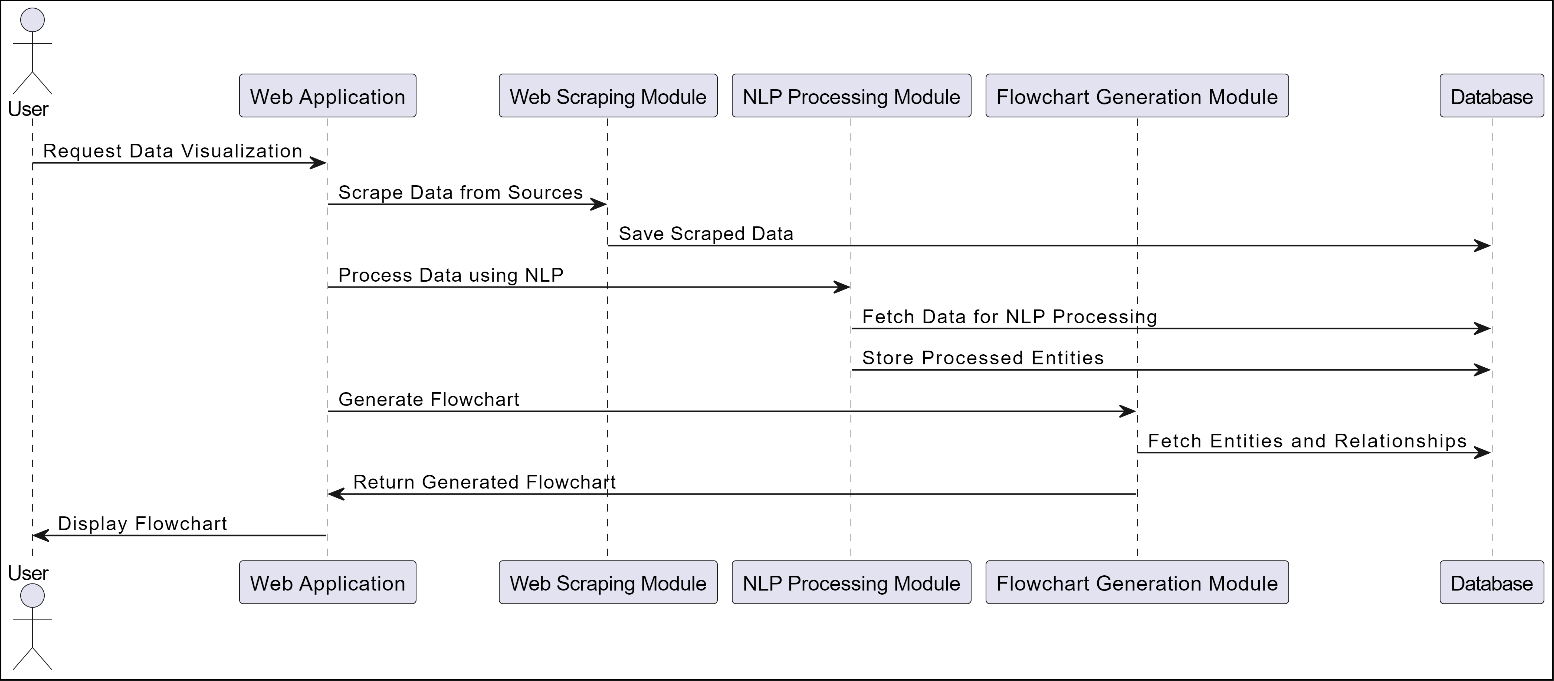
* + 1. **- SEQUENCE DIAGRAM**

Fig.5.4.3: Sequence diagram

* + 1. **- CLASS DIAGRAM**

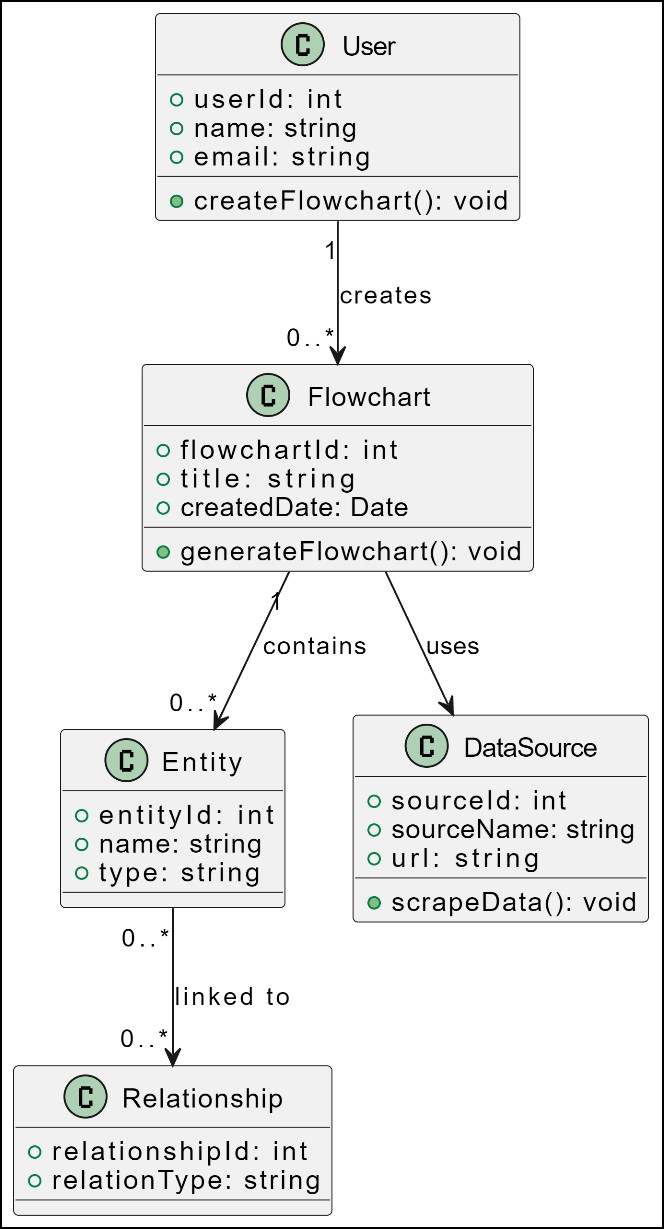
****

Fig.5.4.4: Class diagram

* + 1. **- ACTIVITY DIAGRAM**

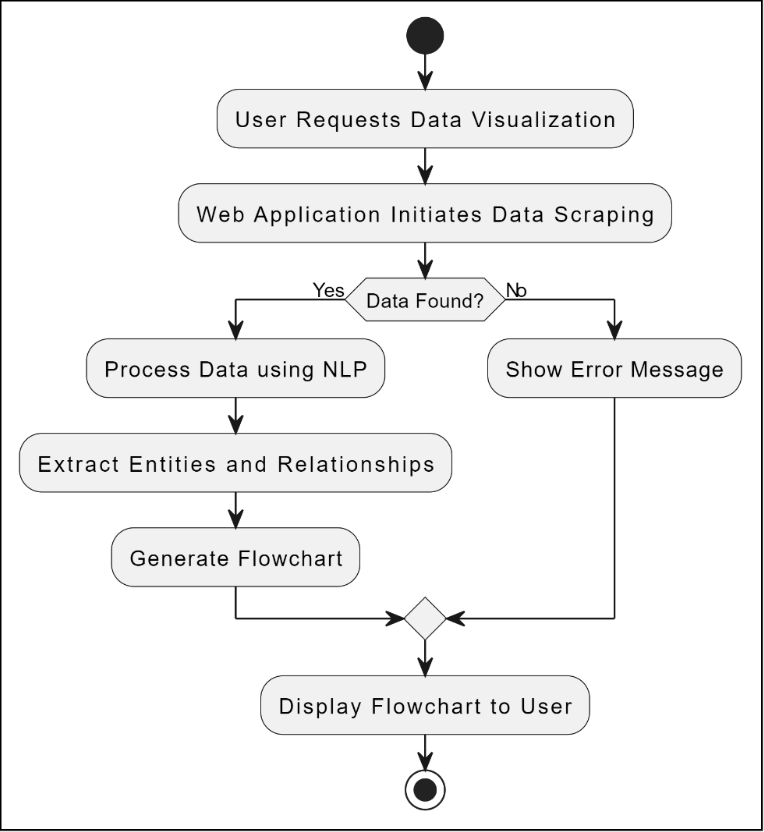
****

Fig.5.4.5: Activity diagram

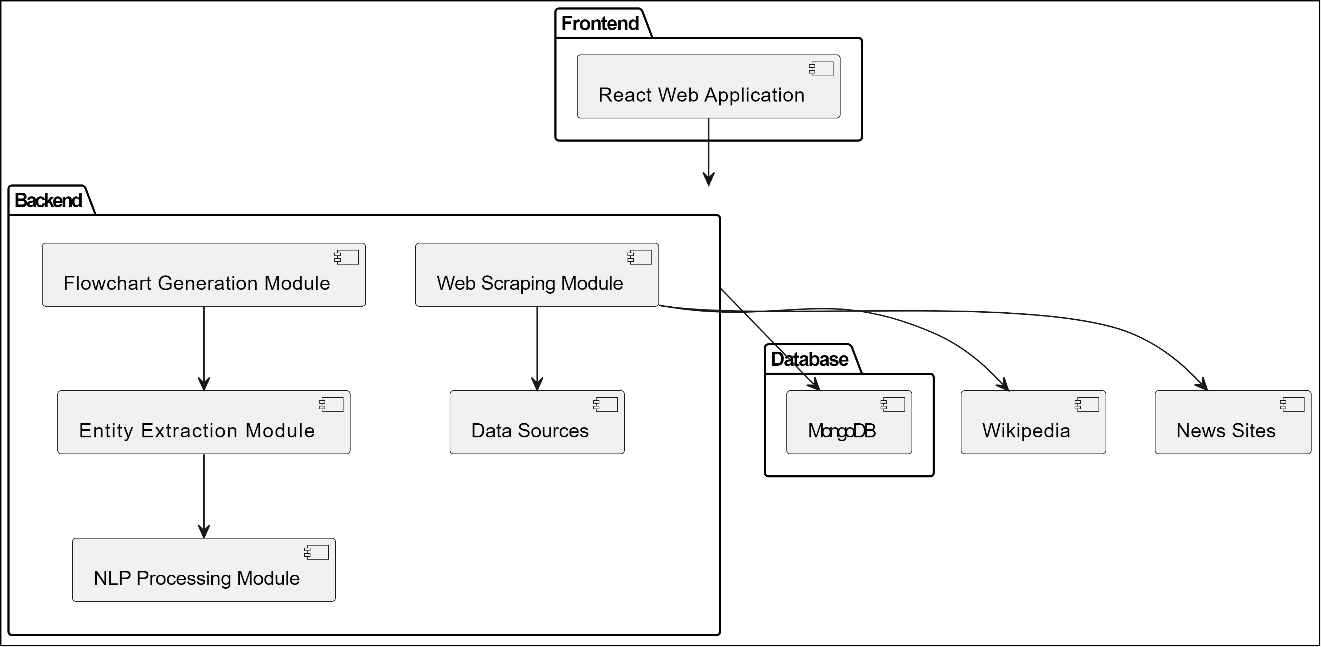
* + 1. **- COMPONENT DIAGRAM**

Fig.5.4.6: Component diagram

**5.4.7 - STATE CHART DIAGRAM**

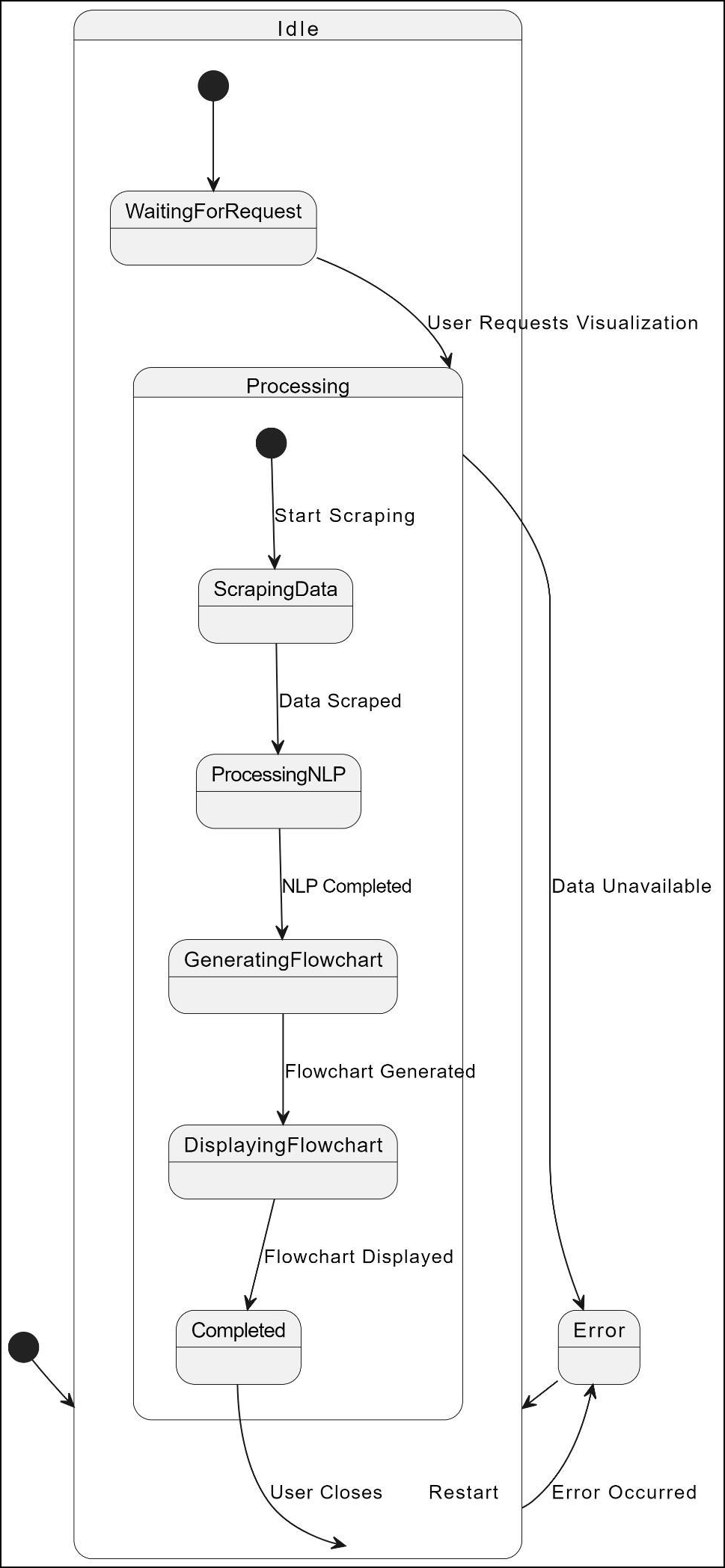
****

Fig.5.4.7: State chart diagram

***Chapter 6***

# IMPLEMENTATION DETAILS

## 6.1 - MODULES AND THEIR FUNCTIONALITIES

The project **"Automated Flowchart Visualization for Understanding Complex Relations"** consists of several interconnected modules, each responsible for a specific functionality. Below is an overview of the key modules and their roles in the system:

1. **Web Scraping Module**
   * **Functionality**: This module is responsible for collecting data from various online sources such as Wikipedia, news platforms, and public archives. It uses web scraping tools like SERP and Cheerio to navigate through different web pages and extract relevant information.
   * **Key Features**:
     + Crawl web pages to extract structured and unstructured data.
     + Handle different page structures, links, and formats to gather accurate and diverse data.
     + Filter and clean raw (organic) data to ensure relevance and accuracy for further processing.
2. **Natural Language Processing (NLP) Module**
   * **Functionality**: This module processes the scraped data to identify and classify key entities such as people, places, and events. It uses Named Entity Recognition (NER) techniques with libraries like NLTK to extract meaningful information from the raw text.
   * **Key Features**:
     + Identify and classify entities like names and locations from unstructured data.
     + Analyze the relationships between the extracted entities (e.g., connections between people and events).
     + Perform sentiment analysis or other text-based analyses if necessary to enrich the data.
3. **Entity-Relationship Processing Module**
   * **Functionality**: This module focuses on processing the output from the NLP module to map relationships between entities. The identified connections are then structured to form a meaningful network of relationships, which is essential for generating flowcharts.
   * **Key Features**:
     + Map relationships between entities based on the context extracted from the data.
     + Handle one-to-one, one-to-many, and many-to-many relationships.
     + Ensure the accuracy of connections to provide reliable insights into the relationships between entities.
4. **Flowchart Generation Module**
   * **Functionality**: This module is responsible for transforming the processed entity- relationship data into dynamic flowcharts. It uses tools like Graphviz to create visual diagrams representing the relationships between different entities.
   * **Key Features**:
     + Automatically generate flowcharts based on processed data.
     + Ensure flowchart accuracy by correctly identifying nodes (entities) and edges (relationships).
     + Create interactive flowcharts with reference links to provide additional context to external resources.
5. **Interactive Web Interface Module**
   * **Functionality**: This module provides a responsive user interface where users can interact with the generated flowcharts. Built using React.Js, the web interface allows users to explore relationships, click on nodes to access external resources, and navigate through the flowcharts seamlessly.
   * **Key Features**:
     + Display flowcharts dynamically based on real-time data.
     + Provide features such as zooming and panning for enhanced user interaction.
     + Offer a clean and intuitive interface that ensures ease of navigation and usability.
6. **Database Management Module**
   * **Functionality**: This module manages the storage and retrieval of scraped data, processed entities, and relationships. It uses local databases to store and organize data efficiently, ensuring smooth flowchart generation and user interaction.
   * **Key Features**:
     + Store scraped and processed data in a structured format.
     + Support querying of data for efficient generation of flowcharts.
     + Manage data updates and deletions to ensure up-to-date information is displayed.
7. **Testing and Validation Module**
   * **Functionality**: This module ensures that the platform is tested thoroughly for bugs and issues. It includes API testing, user interface testing, and database testing. Tools like Postman are used to ensure that all components are functioning as expected.
   * **Key Features**:
     + Perform automated and manual tests to validate the platform’s functionality.
     + Test the accuracy of entity extraction, flowchart generation, and user interaction features.
     + Ensure system stability under load.

***Chapter 7***

# SNAPSHOTS/ GUI

## 7.1 -Search Result:

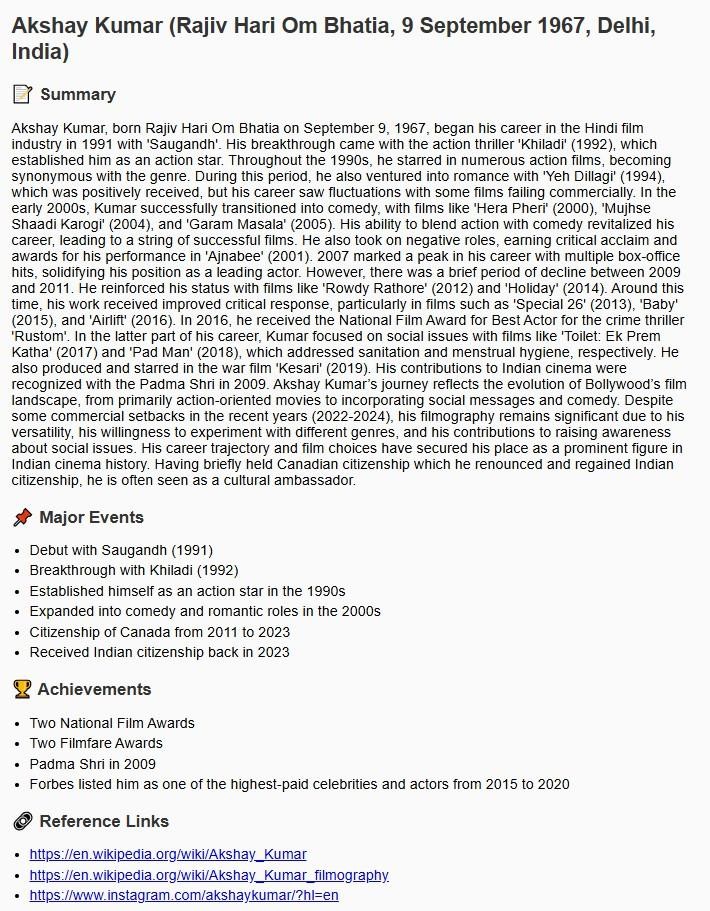
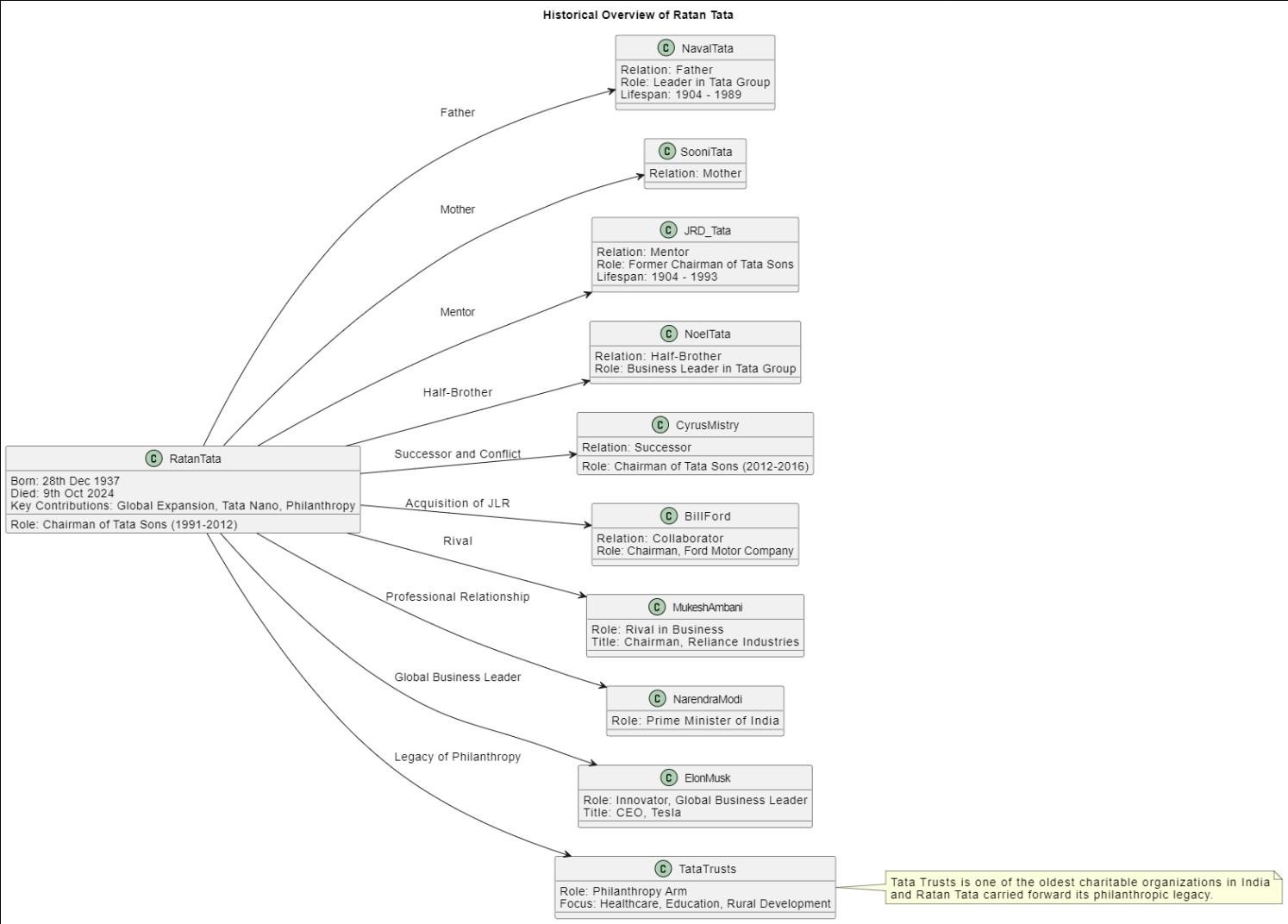
Prompt – Akshay Kumar

Fig.7.1: Search result

## 7.2 -Flowchart Result:

Prompt – Ratan Tata

Fig.7.2: Flowchart result



***Chapter 8***

# CONCLUSION

The **Automated Flowchart Visualization for Understanding Complex Relations** project provides a streamlined approach to navigating vast, unstructured datasets, transforming them into interactive flowcharts. By integrating web scraping, natural language processing (NLP), and dynamic visualization tools like Graphviz, this platform simplifies the exploration of complex relationships between people, events, and locations. The use of modern technologies such as React.Js ensures that the system is responsive, scalable, and user-friendly. This project not only serves as an educational tool but also addresses the growing need for efficient data visualization in various areas like research, media, and education. By automating the extraction and mapping of relationships, users can now interact with data in a more intuitive and engaging way, ultimately leading to faster insights and better decision-making.

The successful implementation of this project highlights the potential of combining multiple fields—web scraping, NLP, and data visualization—to solve real- world problems. Future enhancements could include expanding the system to support more diverse data sources, refining NLP techniques for greater accuracy, and incorporating real-time data processing capabilities.

***Chapter 9***

# BIBLIOGRAPHY

* “Automatic Generation of UML Class Diagrams from Natural Language

Requirements,” in IEEE International Conference on Computer and Information Technology (CIT), 2019.

* “Automated Flowchart Generation from Natural Language Requirements,” in IEEE International Conference on Software Quality, Reliability and Security (QRS), 2020.
* “Framework for Extracting Information from Webpages Using NLP and Web Scraping Techniques,” in 2021 International Conference on Emerging Smart Computing and Informatics (ESCI).
* “Entity Relationship Extraction from Web Text Using Distant Supervision,” in 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA.