Luxor university

Faculty of Computer and Information

Computer Science Department

Smart Tourism Development System

Graduation Project

Under Supervision

DR/Mohamed Fouad

DR/Amany Ashraf

Submitted by

Kariem Abdelmoniem Ahmed

Kirolos Raouf Helmy

Abdelrahman Mohamed

Ahmed Mohamed Nabil

Mohamed Ashraf Mohmed

Mahmoud Mohamed Sharfy

ABSTRACT

The Smart Tourism Development System is a cutting-edge, data-driven platform designed to revolutionize Egypt's tourism industry. By integrating data analytics, machine learning, and cloud technologies, the system addresses core challenges such as inconsistent service quality, fragmented feedback mechanisms, and limited actionable insights for strategic decision-making. Through automated data scraping, sentiment analysis, role-based dashboards, and predictive analytics, the system empowers stakeholders—from government entities to local businesses—to enhance service quality, improve tourist satisfaction, and drive sustainable tourism growth. Developed using Python, Angular, and AWS Cloud Infrastructure, this platform aligns with Egypt's cultural and economic goals, ensuring competitiveness in the global tourism market.

Table of Contents

Content **Page**

List of Figures

Figure **Page**

List of Tables

Table **Page**

Chapter 1

System Overview

Introduction

The Smart Tourism Development System is an innovative data-driven solution designed to enhance the tourism experience in Egypt. By leveraging data analytics, machine learning, and cloud technologies, the system aims to address common challenges faced by the tourism sector, such as inconsistent service quality, fragmented feedback mechanisms, and a lack of actionable insights for decision-makers.

Motivation

Egypt's tourism sector plays a crucial role in the nation's economy but faces several challenges that limit its full potential. The Smart Tourism Development System seeks to address these issues by:

- Improving service quality through data-driven insights.
- Streamlining feedback mechanisms to provide actionable recommendations.
- Supporting stakeholders with tools to benchmark performance against global competitors.
- Promoting sustainable tourism growth while preserving Egypt's cultural heritage.

By transforming tourist feedback into valuable insights, the system empowers decision-makers to enhance visitor satisfaction and ensure Egypt remains a competitive and attractive destination for global travelers.

Problem Statement

Despite its historical and cultural significance, Egypt's tourism industry faces persistent challenges that hinder its growth and global competitiveness:

- **Inadequate infrastructure and services**: Tourists often report dissatisfaction with the quality of facilities and services provided.
- **Poor handling of tourist feedback and complaints**: Feedback mechanisms are often fragmented, making it difficult to address tourists' concerns effectively.
- **Limited actionable insights**: Feedback data is underutilized, resulting in missed opportunities for improvement.
- **Lack of strategic decision-making**: Decisions are often made without leveraging data-driven insights, leading to suboptimal outcomes.

These challenges result in reduced tourist satisfaction and limit Egypt's ability to meet international standards.

Overview

The Smart Tourism Development System is a comprehensive platform designed to modernize Egypt's tourism industry. Key features of the system include:

- Automated Data Scraping: Extracting valuable tourist feedback from platforms like TripAdvisor.
- Sentiment Analysis: Utilizing machine learning algorithms to gauge tourist sentiment and identify common themes in feedback.
- Role-Based Dashboards: Offering customized views and insights tailored to government bodies, tourism operators, and local businesses.
- Predictive Analytics: Providing foresight into potential challenges and opportunities through advanced analytics.

The system is developed using Python for data processing, Angular for an intuitive frontend experience, and deployed on AWS Cloud Infrastructure for scalability and reliability. It aims to:

- Enhance decision-making processes by providing clear, actionable insights.
- Improve tourist satisfaction by addressing pain points.
- Foster sustainable tourism development that aligns with Egypt's cultural and economic goals.

Chapter 2

Related Work

Introduction

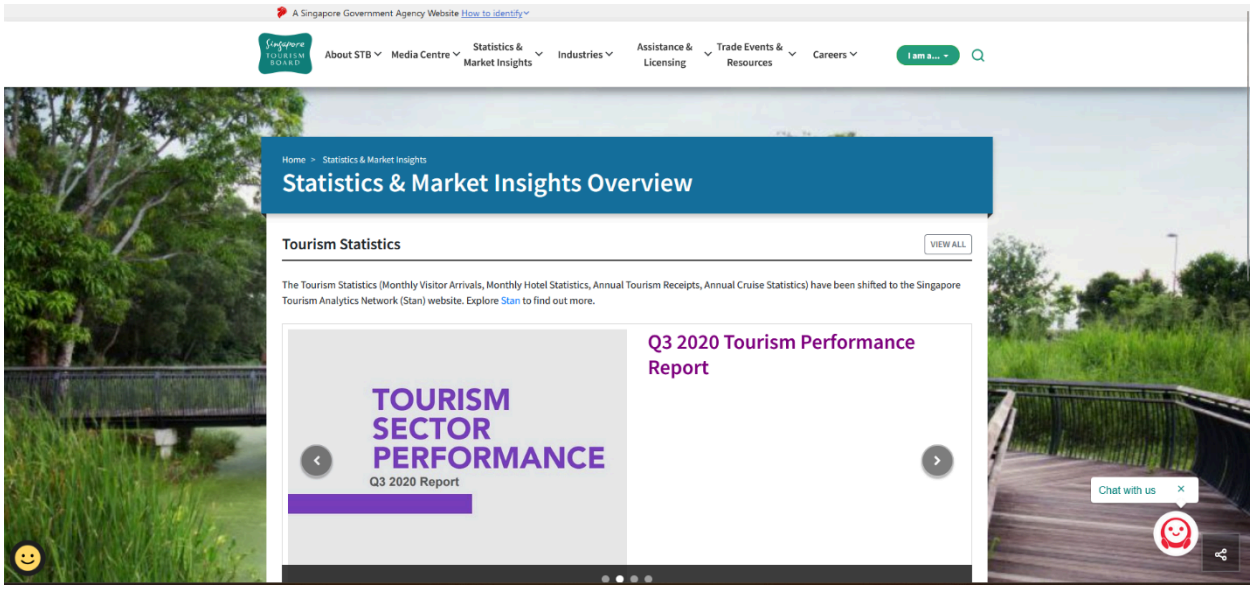
This chapter explores how data-driven strategies have transformed the tourism sector of Singapore through the efforts of the Singapore Tourism Board (STB). By analyzing Singapore's achievements and comparing them to Egypt's current tourism landscape, the chapter identifies opportunities for growth and highlights how the Smart Tourism Development System can address existing challenges. Specific focus areas include visitor arrivals, tourism receipts, infrastructure development, marketing campaigns, and visitor satisfaction. These comparisons provide valuable insights into the potential for innovation and sustainable growth in Egypt's tourism sector.

Objectives

The objectives of this chapter are as follows:

- **Understand Data-Driven Success Stories:** Examine how the Singapore Tourism Board utilized data-driven strategies to transform its tourism sector.
- **Identify Growth Opportunities for Egypt:** Compare Singapore's advancements to Egypt's current tourism landscape to pinpoint areas for improvement.
- **Highlight the Potential of Data Analytics:** Showcase how data analytics, predictive modeling, and sentiment analysis can address challenges in the tourism industry.

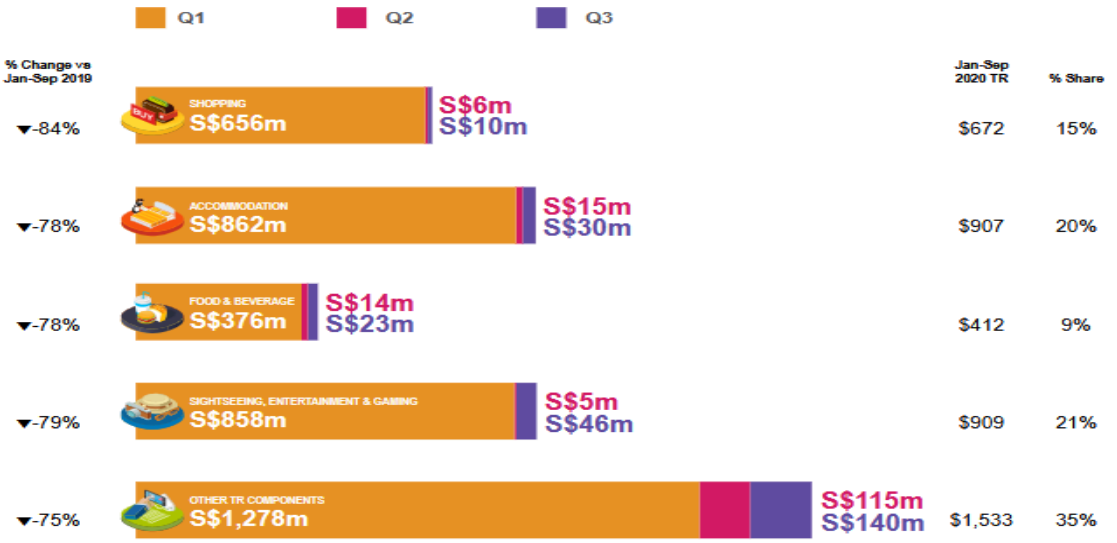
Snippets of the related works (Singapore Tourism Board (STB))

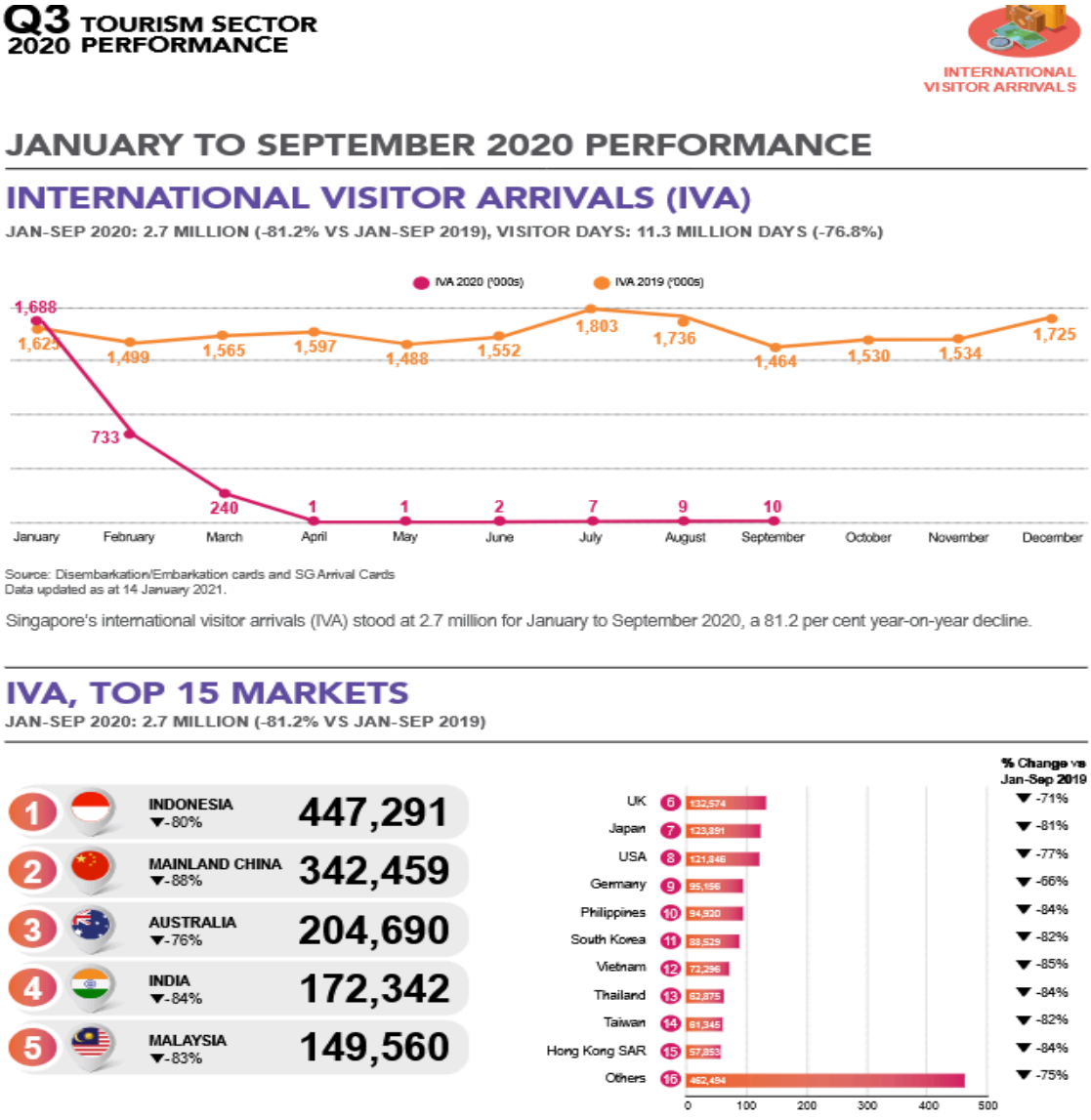


JANUARY TO SEPTEMBER 2020 PERFORMANCE

TOURISM RECEIPTS BY MAJOR COMPONENTS[1]

TOURISM RECEIPTS: S$4.4BILLION (-78.4% VS Jan-Sep 2019)





Inspiration from the Singapore Tourism Board (STB) and Opportunities for Egyptian Tourism

The Singapore Tourism Board (STB) serves as a prime example of how data-driven strategies can revolutionize a nation's tourism sector. Since its establishment in 1964, STB has effectively leveraged insights and

analytics to transform Singapore into a global tourism hub. Comparing this to Egypt's tourism sector highlights opportunities for growth and innovation through the implementation of data-driven strategies in the Smart Tourism Development System.

Visitor Arrivals

Singapore Before Data-Driven Strategies: In the 1960s, Singapore welcomed approximately 91,000 international visitors annually. Tourism relied on traditional promotional methods, which lacked precision and efficiency.

Singapore After Implementing Data Insights: In 2023, Singapore recorded 13.6 million international visitors, reflecting a remarkable growth of over This was achieved through the use of data analytics, predictive modeling, and targeted marketing campaigns.

Egyptian Context and Opportunities: Egypt receives millions of visitors annually, attracted by its rich cultural heritage and historical landmarks. However, the lack of targeted marketing and predictive analytics limits growth potential. By employing market segmentation and predictive analytics similar to Singapore, Egypt could:

- Attract more international tourists.
- Optimize seasonal campaigns to align with visitor preferences.

Tourism Receipts

Singapore Before Data-Driven Strategies: Tourism receipts were minimal due to the lack of understanding of visitor spending habits and behaviors.
Singapore After Implementing Data Insights: By 2023, tourism receipts had reached $24.5 to $26.0 billion, even recovering to 88-94% of pre-pandemic levels despite global challenges.
Egyptian Context and Opportunities: Egypt's tourism receipts are substantial but often not optimized. With advanced analytics, Egypt could:

- Identify high-value visitors and tailor experiences to increase spending.
- Diversify offerings, such as luxury, eco-tourism, and cultural tourism, to cater to niche markets.

Infrastructure Development

Singapore Before Data-Driven Strategies: Tourism infrastructure development was largely ad-hoc and reactive, leading to inefficiencies and missed opportunities.

Singapore After Implementing Data Insights: Data-driven planning facilitated the creation of iconic attractions such as Marina Bay Sands, Gardens by the Bay, and Sentosa Island, which attract millions of visitors annually.

Egyptian Context and Opportunities: Egypt has world-renowned landmarks but lacks the modern infrastructure to enhance the visitor experience. By using demand forecasting and experience mapping, Egypt could:

- Develop transportation and amenities that streamline access to sites like the Pyramids and Luxor.
- Introduce innovative attractions to complement historical sites.

Marketing Campaigns

Singapore Before Data-Driven Strategies: Marketing campaigns lacked focus and relied on generic messaging, which limited their impact.

Singapore After Implementing Data Insights: Campaigns like “Passion Made Possible” used social media sentiment analysis and online behavior data to target specific demographics effectively.

Egyptian Context and Opportunities: Egypt's marketing efforts often focus on traditional channels and general messaging. By leveraging social media analytics and performance metrics, Egypt could:

- Tailor campaigns for different demographics, such as adventure travelers or cultural enthusiasts.
- Monitor real-time feedback to adjust messaging dynamically.

Visitor Satisfaction

Singapore Before Data-Driven Strategies: Visitor feedback was anecdotal and lacked actionable insights, leading to challenges in addressing dissatisfaction.

Singapore After Implementing Data Insights: Tools like ReviewPro's Global Review Index™ (GRI™) helped STB achieve high satisfaction rates by addressing pain points proactively.

Egyptian Context and Opportunities: While Egypt receives significant tourist feedback, a structured system to analyze and act on this data is missing. By incorporating sentiment analysis tools, Egypt could:

- Identify and resolve common pain points, such as overcrowding or insufficient facilities.
- Enhance overall satisfaction to encourage repeat visits and positive word-of-mouth.

Conclusion

The Singapore Tourism Board's success underscores the transformative power of data-driven strategies in the tourism industry. Comparing this with Egypt's current landscape reveals significant opportunities for improvement. By drawing inspiration from STB's approach, the Smart Tourism Development System aims to:

- Utilize advanced data analytics and predictive modeling.
- Identify weaknesses in the tourism sector.
- Enhance visitor satisfaction and infrastructure development.
- Boost marketing effectiveness and revenue generation.

Through these measures, Egypt's tourism sector can achieve sustainable growth, solidifying its position as a top global destination.

Chapter 3

Domain Analysis and Technique

Introduction

Chapter 3 delves into the technical and analytical framework of the Smart Tourism Development System, focusing on domain analysis, identified risks, constraints, feasibility, and quality assurance. This chapter outlines the system's operational domain, highlighting key components like data collection, processing, and visualization. Additionally, it evaluates the project's viability, identifies potential challenges, and presents a detailed quality assurance plan to ensure the system meets stakeholder expectations.

Objective

The objectives of Chapter 3 are to:

1. Define the operational scope and techniques relevant to the tourism analytics domain.

2. Highlight risks and constraints, emphasizing mitigation strategies to ensure smooth implementation.

3. Conduct a detailed feasibility study to evaluate the system's financial, technical, operational, and regulatory aspects.

4. Present the quality assurance plan to ensure system reliability, performance, and stakeholder satisfaction.

5. Describe the tools and techniques used for data collection, processing, and visualization to achieve the project’s goals.

3.1 Domain Analysis

The **Smart Tourism Development System** operates within the **tourism analytics domain**, focusing on data-driven decision-making to address challenges in **tourist satisfaction, service quality, and infrastructure management**. The system leverages **data analytics, machine learning, and cloud technologies** to gather insights from platforms like **TripAdvisor**.

Key Aspects of the Domain:

- **Data Collection:** Automated scraping of user-generated reviews from tourism platforms.
- **Data Processing:** Sentiment analysis and trend identification using NLP techniques.
- **Data Visualization:** Role-based dashboards for actionable insights.
- **Decision Support:** Data-driven recommendations for stakeholders to improve tourism services and infrastructure.

3.2 Risks

Table 1Risks Table

Risk	Effects	Priority	Strategy
Data Inconsistency or Inaccuracy	Incorrect insights may lead to flawed decision-making.	High	Regular data validation and cleaning processes.
Integration Challenges with Third-Party Platforms	Integration failures may disrupt data collection workflows.	Medium	Use standardized APIs and robust error-handling mechanisms.

Risk of downtime without updates	Ensures consistent functionality	High	Regular system maintenance
Unauthorized data access	Prevents data breaches	High	Secure data handling
Overloaded infrastructure	Accommodates growing user base	Medium	Scalable architecture design
Incorrect sentiment interpretation	Provides reliable insights	Medium	Accurate sentiment analysis

3.3 Constrains

3.3.1. Technical Constraints:

- Data Source Limitations: Access to review data depends on third-party platforms like TripAdvisor, which may have API restrictions or usage limits.
- Real-Time Processing: Ensuring efficient real-time data scraping and analysis without overloading system resources.

- System Scalability: Infrastructure must handle increasing data volumes and concurrent users efficiently.

3.3.2. Financial Constraints:

- Cloud Service Costs: Ongoing expenses for AWS services (RDS, EC2, Lambda) may increase with higher usage and data volumes.
- Maintenance Costs: Continuous support, bug fixes, and infrastructure updates require a sustained budget.

3.4 Project plan

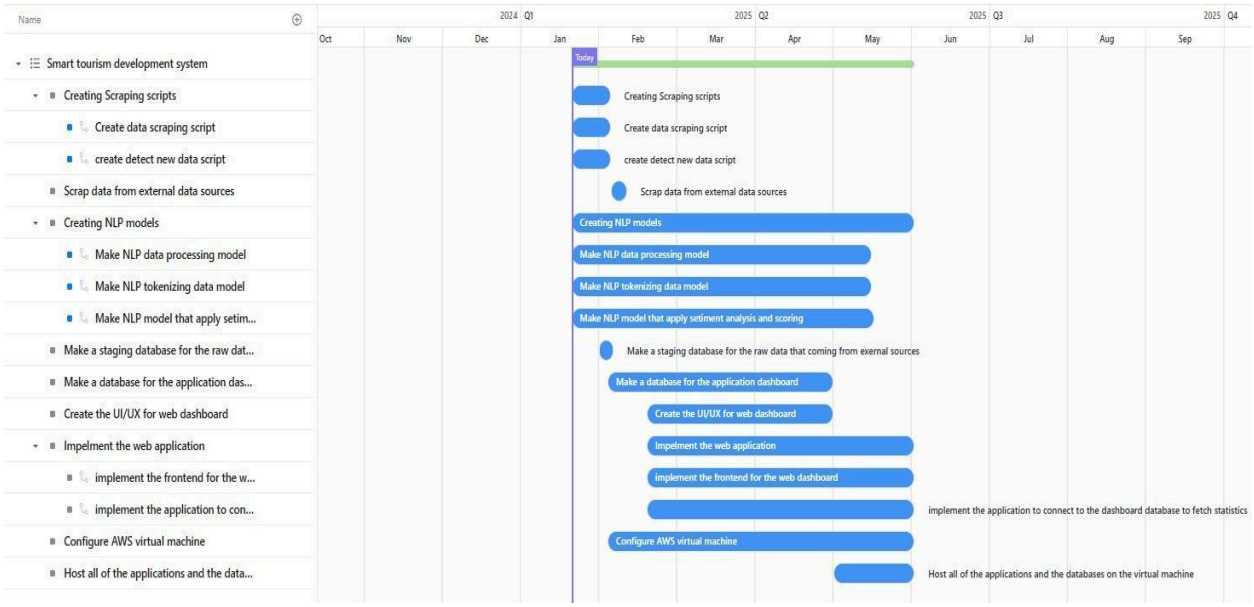


Figure 1 Project Plan

Name	Assignee	Start date	Due date	Priority	Status	Comments
Creating Scraping scripts 2		Today	Feb 4	Urgent	TO DO	
Create data scraping script		Today	Feb 4	Urgent	IN PROGRE...	
create detect new data script		Today	Feb 4	Urgent	IN PROGRE...	
Scrap data from external data sources		Feb 5	Feb 10	Normal	TO DO	
Creating NLP models 3		Today	Jun 1	High	TO DO	
Make NLP data processing model		Today	May 15	High	IN PROGRE...	
Make NLP tokenizing data model		Today	May 15	High	IN PROGRE...	
Make NLP model that apply setiment analysis and sco...		Today, 12am	May 16, 12am	High	IN PROGRE...	
Make a staging database for the raw data that coming from e...		Feb 1	Feb 5	High	TO DO	
Create the UI/UX for web dashboard		Feb 18	Apr 30	Normal	TO DO	
Make a database for the application dashboard		Feb 4	Apr 30	Normal	TO DO	
Impelment the web application 2		Feb 18	Jun 1	Normal	TO DO	
implement the frontend for the web dashboard		Feb 18	Jun 1	Normal	TO DO	
implement the application to connect to the dashboard d...		Feb 18	Jun 1	Normal	TO DO	
Configure AWS virtual machine		Feb 4	Jun 1	Normal	TO DO	
Host all of the applications and the databases on the virtual ...		May 1	Jun 1	Normal	TO DO	

3.5 Feasibility Study

3.5.1. *Introduction*

- **Background:**
  The tourism industry faces challenges in understanding and meeting the expectations of diverse tourists. A data-driven approach is necessary to improve decision-making and enhance the overall tourist experience.
- **Scope:**
  The system will focus on data collection, processing, visualization, and decision support for tourism stakeholders.
- **Methodology:**

Analysis of market demand, technical resources, financial viability, and operational feasibility.

3.5.2 *Market Analysis*

- **Industry Overview:**
  The global tourism industry is increasingly leveraging technology for insights. Digital tools for analytics have become essential to staying competitive.
- **Target Market:**
  Tourism boards, service providers, and policymakers seeking to improve tourist satisfaction and resource management.
- **Competition Analysis:**
  Few existing solutions offer an end-to-end analytics platform. The integration of sentiment analysis and role-based dashboards is a competitive advantage.
- **Demand Forecast:**

Growing demand for analytics in tourism, with increased focus on data-driven strategies.

3.5.3 *Technical Feasibility*

- **Project Requirements:**
  Data Collection: Automated web scraping tools for platforms like TripAdvisor.
  Data Processing: NLP frameworks for sentiment analysis and trend identification.
  Data Visualization: Tools like Power BI or Front-End for creating dashboards.
  Cloud Infrastructure: AWS or Google Cloud for scalability.
- **Infrastructure Availability:**
  Cloud technologies and existing NLP libraries make implementation feasible.
- **Skills and Expertise:**
  Requires expertise in machine learning, cloud computing, and dashboard design.

3.5.4 *Legal and Regulatory Feasibility*

- **Regulatory Requirements:**
  Compliance with data privacy laws in Egypt.
- **Legal Considerations:**
  Ensure ethical data scraping practices and secure handling of user-generated content.

3.5.5 *Operational Feasibility*

- **Resources Availability:**

Skilled professionals available in data science, software development, NLP, and tourism analytics.

- **Management Structure:**
  Led by a project manager and supervisor dr. Mohamed Fuaad
- **Timeframe:**
  Estimated development time of 4-6 months.

3.5.6 **Risk Assessment**

- **Potential Risks:**
  Financial Risks: Higher-than-expected operational costs.
  Operational Risks: Difficulty in maintaining data scraping tools.
  Market Risks: Slow adoption by stakeholders.
  Legal Risks: Non-compliance with data privacy regulations.
- **Mitigation Strategies:**
  Conduct market validation to ensure demand.
  Implement robust compliance protocols.
  Allocate contingency funds for unforeseen expenses.

3.6 Quality Assurance Plan

:Testing Methodology .1

.**Unit Testing:** Validate individual modules and functions •

Integration Testing: Ensure seamless communication between •

.backend, frontend, and database

System Testing: End-to-end validation of workflows and data •

.pipelines

Performance Testing: Assess system scalability and response •

.times

:Documentation and Reviews •

.Regular code reviews to maintain clean, efficient code •

.Maintain up-to-date technical documentation and user manuals •

Conduct stakeholder feedback sessions to ensure alignment with •

.expectations

System Requirements

:Hardware Requirements .1

.Server Infrastructure: AWS EC2 for hosting backend services •

.Database Server: AWS RDS for scalable data storage •

Client Devices: Standard laptops or desktops with modern web •

.browsers

:Software Requirements .2

.Backend: Python, MongoDB, AWS RDS •

.Frontend: Streamlit •

.Data Analytics: Pandas, NLTK •

3.8 Techniques and tools

1. **Data Collection Techniques:**

- **Web Scraping:** Python scripts using libraries like **BeautifulSoup (bs4)** and **Selenium** to collect data from **Booking.com**.
- **API Integration:** Standard APIs to pull structured data when available.

2. Data Processing Techniques:

- **Sentiment Analysis:** NLP libraries like **NLTK** to classify reviews into positive, neutral, or negative.
- **Data Cleaning and Transformation: Pandas** for organizing and preparing data for analysis.
- **Predictive Analytics:** Machine Learning models to forecast trends and identify potential issues.

3. Visualization Techniques:

- **Interactive Dashboards: Streamlit** for building user-friendly and efficient dashboards for data visualization and interaction.
- **Graph Libraries: Matplotlib**, **Seaborn**, **Plotly** and **Dash** for creating dynamic and visually appealing charts and graphs within Streamlit dashboards.

Table 2 Tools Table **Category**	**Techniques/Tools**	**Description**
Description	Web Scraping: BeautifulSoup, Selenium	Web scraping involves extracting data from websites. BeautifulSoup parses HTML and XML documents to retrieve relevant data, while Selenium automates browser actions for dynamic sites.
Automation	Automation Using Apache Airflow	maintain a robust, scalable pipeline, we integrated **Apache Airflow** to automate and orchestrate the
	API Integration	APIs provide structured access to data from external platforms. Using standard APIs ensures

		efficient, real-time data extraction directly from trusted sources.
Data Processing	Sentiment Analysis: NLTK	Natural Language Processing (NLP) tools analyze review text to determine whether the sentiment expressed is positive, neutral, or negative, aiding in understanding customer opinions.
	Data Cleaning and Transformation: Pandas	Pandas is used for data manipulation, cleaning, and transformation, ensuring the dataset is structured, consistent, and ready for analysis.
	Predictive Analytics: Machine Learning models	Machine learning techniques are used to identify patterns, predict future trends, and provide actionable insights for strategic planning and decision-making.
Visualization		Streamlit and dash is a Python-based tool for creating

	Interactive Dashboards: Streamlit	intuitive, interactive web applications and dashboards, enabling users to explore and analyze data effectively.
	Graph Libraries:Stremlit, Matplotlib, Seaborn, Plotly,Dash	These libraries create static and interactive visualizations, such as line graphs, bar charts, and heatmaps, to convey complex data insights in a clear format.

Chapter Conclusion

This chapter provided a comprehensive analysis of the domain, highlighting the system's data-driven approach to addressing challenges in the tourism sector. Key risks, constraints, and feasibility considerations were discussed to ensure the project's practicality and alignment with stakeholder needs. The quality assurance plan,

combined with advanced techniques and tools, lays a solid foundation for the successful development and deployment of the Smart Tourism Development System. This sets the stage for the next phase, focusing on system design and implementation.

Chapter 4

Proposed System & Methodology

Introduction

Chapter 4 presents the proposed system and methodology for the Smart Tourism Development System. This chapter focuses on the detailed description of use cases, system architecture, and design elements, which form the backbone of the system's operations. It also outlines key processes such as scraping reviews, comments analysis, and generating insights, as well as the technical workflows required to support these functions. The use case scenarios describe the key interactions between the system and its users, while the design diagrams provide a visual representation of system components and their interactions.

Objective

The objective of Chapter 4 is to:

1. Present detailed use case scenarios for the core functions of the Smart Tourism Development System, including data scraping, sentiment analysis, and presenting insights.

2. Define the system architecture, outlining how various components interact to achieve the system's goals. 3. Provide class diagrams, activity diagrams, and data flow diagrams to visualize system processes and data flow.

4. Explain the design and database schema, which support the system’s functionalities and ensure efficient data management.

4.1 System Use-Cases

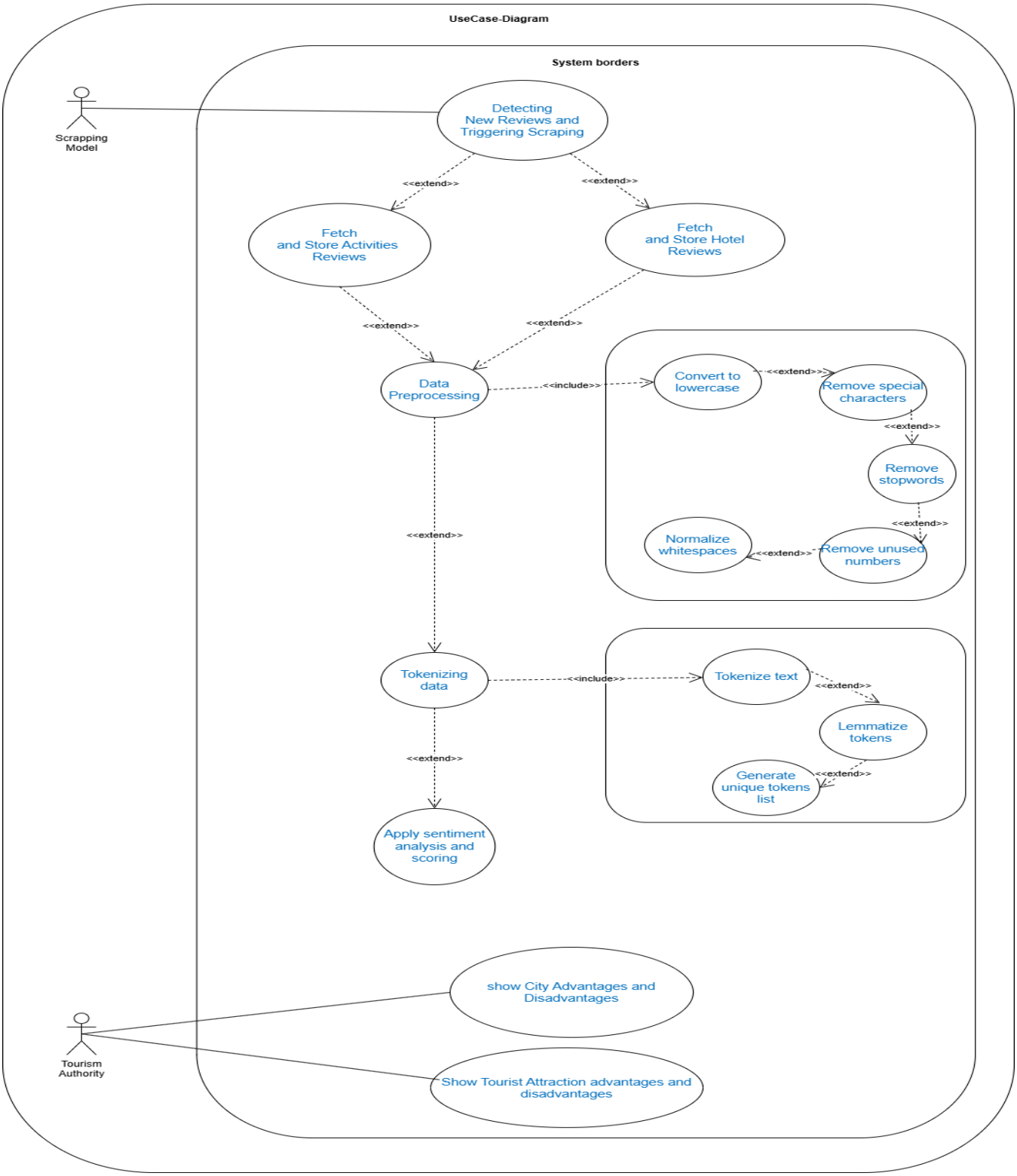


Figure 2 Use Case

4.2 Use Case Description (Use case scenario)

Use Case ID:	UC001
Use Case Title	Detecting New Reviews and Triggering Scraping
Version	1.0
Actor	Scraping Model
Description	This use case describes the process of checking if there is new review inserted or not Starts with: The model send request into database to get last review stored Ends with: Pull the trigger to run the scraping scripts .
Precondition	Database isn't empty
Post Conditions	The trigger status of the scraping scripts is updated based on the detection of new reviews.
Main Flow	1. The model retrieves the last stored review from the database. 2. The model retrieves the latest review from the website. 3. The model compares the database record with the latest review on the website to identify discrepancies.
Alternative Flow	3a. If new reviews are detected, the model triggers the execution of the scraping scripts to fetch the latest reviews. 3b. If no new reviews are detected, the model terminates the process without triggering the scripts.

Table 3 Use Case 1 Scenario

Use Case ID:	UC002
Use Case Title	Fetching and Storing Hotel Reviews
Version	1.0
Actor	Scraping Model
Description	*This use case describes the process of fetching and storing reviews related to hotels.* *Starts with: The scraping scripts are triggered.* *Ends with: The reviews are stored in the database.*
Precondition	New reviews inserted into website
Post Conditions	*New reviews stored in the database*
Main Flow	*1. Model send request into website to fetch all reviews.* *2. Model retrieves all reviews.* *3. the model stores the new reviews into the database.*

Alternative Flow Table 4 Use Case 2 Scenario	*1a. request failed resend request again and store in the log file*

Use Case ID:	UC003
Use Case Title	Fetching and Storing Activities Reviews
Version	1.0
Actor	Scraping Model
Description	*This use case describes the process of fetching and storing reviews for activities.* *Starts with: The scraping scripts are triggered.* *Ends with: Activity reviews are stored in the database.*
Precondition	New reviews inserted into website
Post Conditions	*New reviews stored in the database*

Main Flow	*1. Model send request into website to fetch all reviews.* *2. Model retrieves all reviews.* *3. The model stores the new reviews into the database.*
Alternative Flow	*1a. request failed resend request again and store in the log file*

Table 5 Use Case 3 Scenario

Use Case ID:	UC004
Use Case Title	Data Preprocessing
Version	1.0
Actor	NLP Model
Description	*This use case describes the preprocessing of scraped data to prepare it for analysis.* *Starts with: The NLP model receives raw scraped data.* *Ends with: Preprocessed data is ready for tokenization..*
Precondition	Raw review data is available in the database.
Post Conditions	*The raw data is cleaned and ready for tokenization.*

Main Flow	1. *The model converts the text to lowercase.* 2. *The model removes special characters from the text.* 3. *The model removes stop words.* 4. *The model removes unused numbers from the text.* 5. *The model normalizes whitespace in the text.*
Alternative Flow Table 6 Use Case 4 Scenario	*2a. If the text contains unremovable characters, they are replaced with placeholders.* *3a. If stop words are not identified correctly, the system logs the error for manual intervention.*
Use Case ID:	UC005
Use Case Title	Tokenizing Data
Version	1.0
Actor	NLP Model
Description	*This use case describes the process of splitting the preprocessed data into tokens for analysis.* *Starts with: Preprocessed text data is provided to the NLP model.* *Ends with: Tokens and unique tokens list are generated.*
Precondition	Preprocessed data is available.

Post Conditions	*Tokenized and lemmatized data is ready for sentiment analysis.*
Main Flow	1. *The model tokenizes the text into individual words.* 2. *The model lemmatizes each token.* 3. *The model generates a unique tokens list for analysis.*
Alternative Flow	*2a. If lemmatization fails, the system retries using a default rule-based approach.* *3a. If duplicate tokens are not removed, the system logs the issue for correction.*

Table 7 Use Case 5 Scenario

Use Case ID:	UC006
Use Case Title	Apply Sentiment Analysis and Scoring
Version	1.0
Actor	NLP Model
Description	*This use case describes the process of applying sentiment analysis to reviews and generating sentiment scores.* *Starts with: Tokenized and lemmatized data is provided to the NLP model.* *Ends with: Sentiment scores are stored for each review.*

Precondition	Tokenized data is available for sentiment analysis.
Post Conditions	*Sentiment scores are updated in the database for each review..*
Main Flow	1. *The NLP model analyzes the sentiment of each review.* 2. *The model assigns a sentiment score to each review.* 3. *The sentiment scores are stored in the database*
Alternative Flow	*1a. If sentiment analysis fails, the system retries*

Table 8 Use Case 6 Scenario

Use Case ID:	UC007
Use Case Title	Show City Advantages and Disadvantages
Version	1.0
Actor	Tourism Authority
Description	*This use case describes the process of displaying city-level advantages and disadvantages based on analyzed reviews.* *Starts with: Sentiment scores are aggregated at the city level.* *Ends with: Insights on city advantages and disadvantages are displayed to the Tourism Authority.*

Precondition	Sentiment scores for all reviews are available.
Post Conditions	*Tourism authorities gain insights into city performance.*
Main Flow	1. *The user navigates to the dashboard page. (User Action)* 2. *The website displays a list of cities on the dashboard. (System Response)* 3. *The user selects a city from the list and clicks on it. (User Action)* 4. *The website sends get request to the backend server to fetch city related data. (System Action)* 5. *The backend server processes the request and responds to the request with city-related data in JSON format. (System Action)* 6. *The website receives the data. (System Response)* 7. *The website navigates the user to the city statistics page. (System Response)* 8. *The website displays the advantages and disadvantages of the selected city. (System Response)*
Alternative Flow Table 9 Use Case 7 Scenario	*5a. The backend server failed to processes the request. (System Action)* *5b. The backend server send message "Something went wrong" (System response)* *5c. The system returns the user to step 3. (System Action)*
Use Case ID:	UC008
Use Case Title	Show Tourist Attraction advantages and disadvantages
Version	1.0
Actor	Tourism Authority

Description	*This use case describes the process of displaying advantages and disadvantages for specific tourist attractions.* *Starts with: Sentiment scores are aggregated for tourist attractions.* *Ends with: Insights on tourist attraction advantages and disadvantages are displayed to the Tourism Authority.*
Precondition	Sentiment scores for all reviews are available.
Post Conditions	*Tourism authorities gain insights into tourist attraction performance.*
Main Flow	1. ***User Action**: The user navigates to the dashboard and selects a city.* 2. ***System Action**: The website fetches city-related data from the backend.* 3. ***System Response**: The website displays the city statistics page with tourist attractions listed.* 4. ***User Action**: The user selects a tourist attraction.* 5. ***System Action**: The website fetches tourist attraction data from the backend.* 6. ***System Response**: The website displays the tourist attraction statistics page.* 7. ***System Response**: Advantages and disadvantages of the tourist attraction are displayed.*
Alternative Flow	*5a. The backend server failed to processes the request. (System Action)* *5b. The backend server send message "Something went wrong" (System response)* *5c. The system returns the user to step 3. (System Action)*

4.3 System Architecture

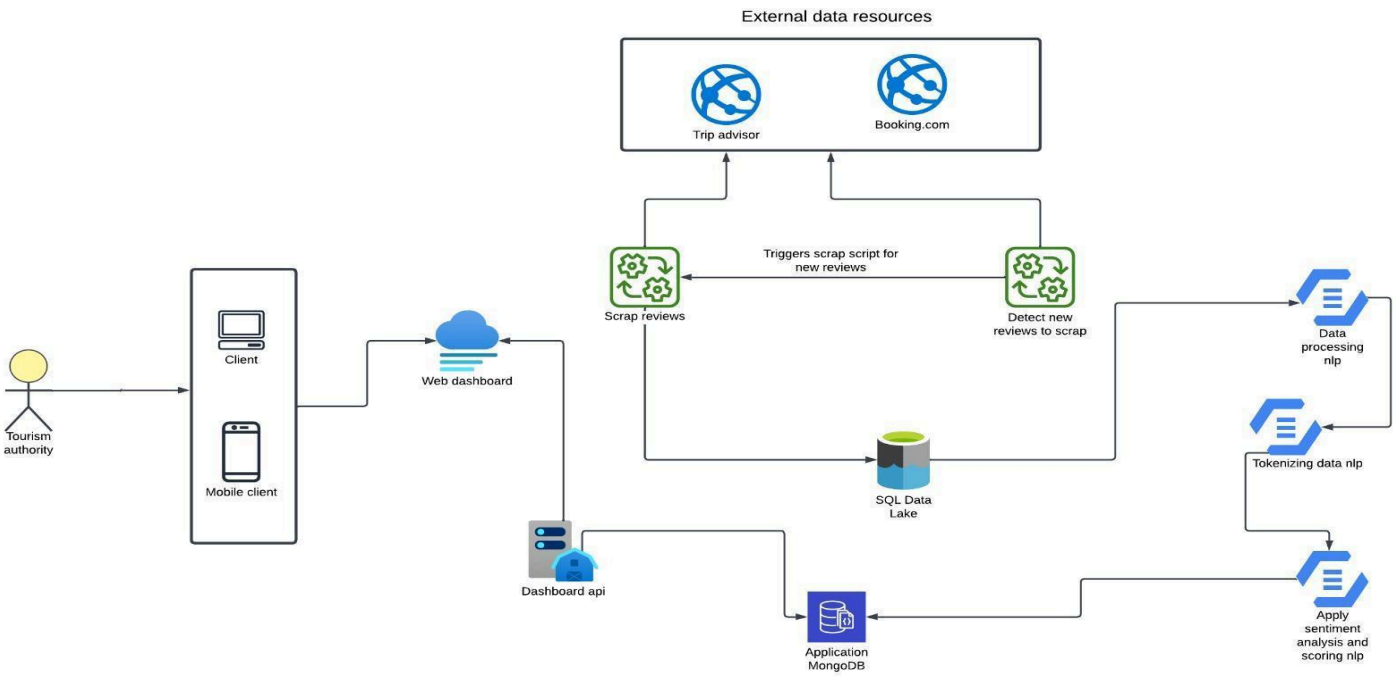


Figure 3 System Architecture

The **Smart Tourism Development System Architecture** is designed to streamline the collection, processing, and delivery of actionable insights from tourist feedback.

1. **Data Collection**: Tourist reviews are collected from external platforms like TripAdvisor and Booking.com using automated scraping scripts.
2. **Data Storage**: Extracted reviews are stored in an SQL Data Lake, serving as the central repository for raw data.
3. **Data Processing**: Reviews undergo preprocessing, tokenization, and sentiment analysis using NLP techniques to generate insights and feedback scores.
4. **Insights Delivery**: Processed data is stored in MongoDB for efficient retrieval and scalability.
5. **Dashboard Integration**: Insights are made accessible via a role-based dashboard, which is connected through an API and supports both web and mobile clients.
6. **Real-Time Updates**: The system continuously detects and processes new reviews to ensure up-to-date information.

7. **End-User Access**: Tourism authorities can access these insights through visualizations and reports for data-driven decision-making.

4.4 Analysis Class

4.4.1 Activity Diagram

4.4.1.1 Scraping-Model Activity

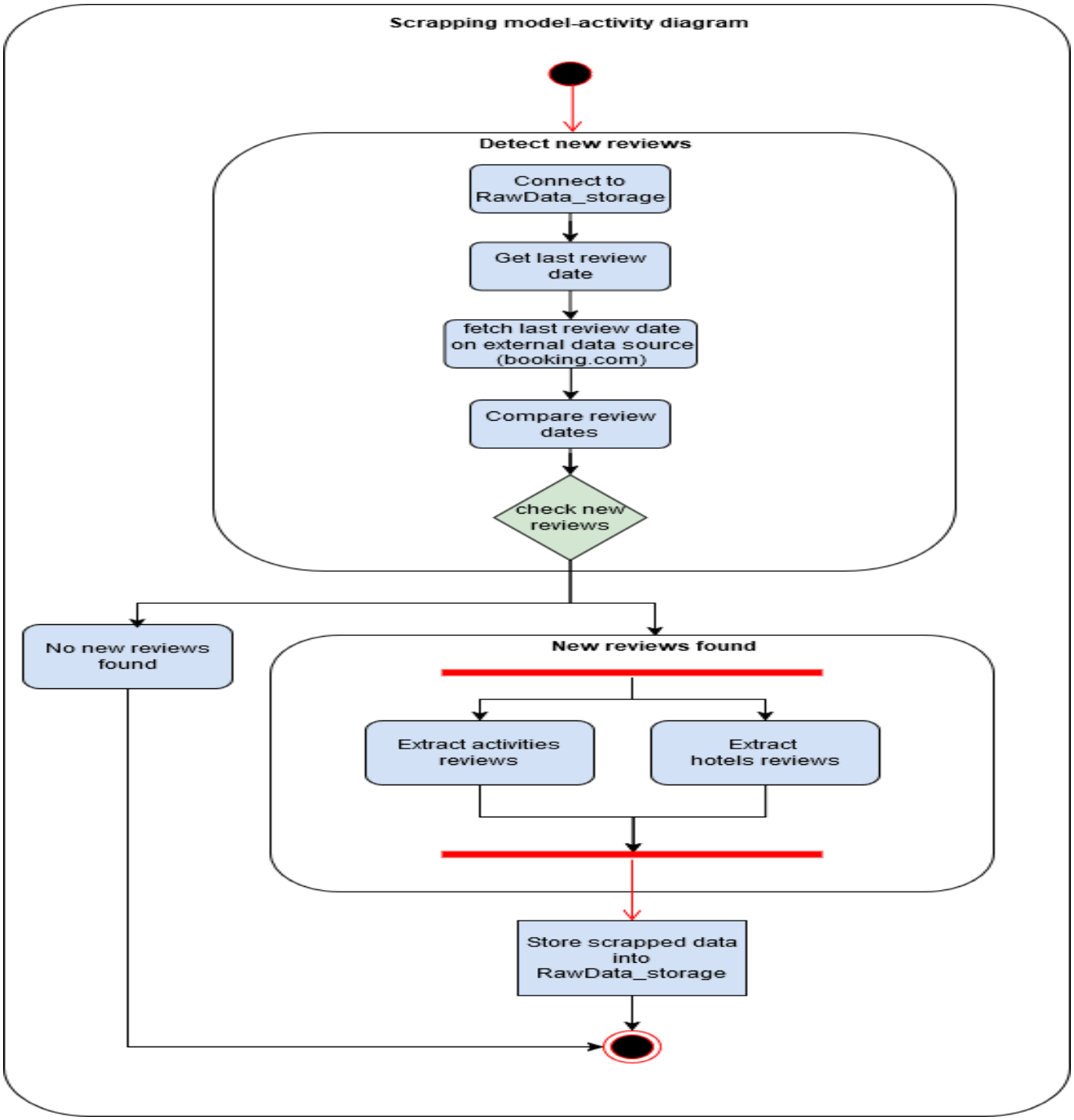


Figure 4 Scraping-Model Activity

The Scraping Model Activity Diagram outlines the process of detecting and extracting new reviews from external sources like Booking.com.

1. Detect New Reviews:

- The system connects to the RawData_storage to retrieve the date of the latest stored review.
- It fetches the latest review date from the external source and compares it with the stored date.

2. Decision Point:
    - If no new reviews are found, the process ends.
    - If new reviews are detected, the system proceeds to extract them.
3. Data Extraction:
    - Reviews are categorized into two types: activity reviews and hotel reviews.
    - The system extracts both types of reviews from the external source.
4. Data Storage:
    - Extracted reviews are stored in the RawData_storage for further processing.

4.4.1.2 NLP-Model Activity

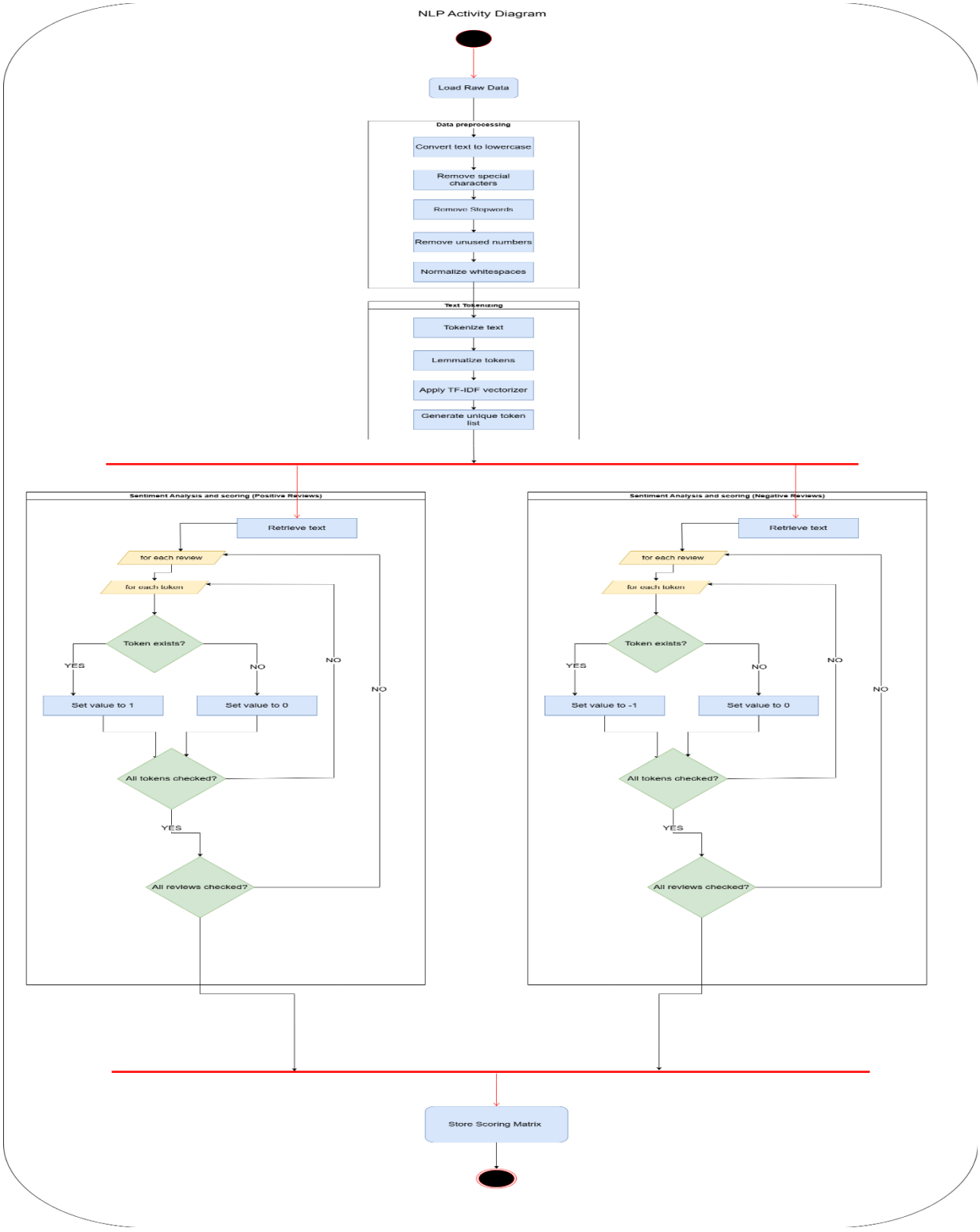


Figure 5 NLP-Model Activity

The NLP Activity Diagram outlines the process of analyzing and scoring reviews through natural language processing (NLP):

1. Data Preparation:
    - Raw data is loaded and preprocessed by converting text to lowercase, removing special characters and numbers, eliminating stop words, and normalizing whitespace.
    - The text is tokenized, lemmatized, and vectorized using techniques like TF-IDF, generating tokens for analysis.
2. Sentiment Scoring:
    - For each review, the system checks the presence of tokens and assigns initial values.
    - Tokens are verified for relevance using predefined criteria and scoring mechanisms.
3. Review Analysis:
    - Tokens and reviews are iteratively checked and evaluated for inclusion in the scoring matrix.
4. Storing Results:
    - The final scoring matrix is stored for further processing and integration into dashboards or reports.

4.4.2 Context Diagram

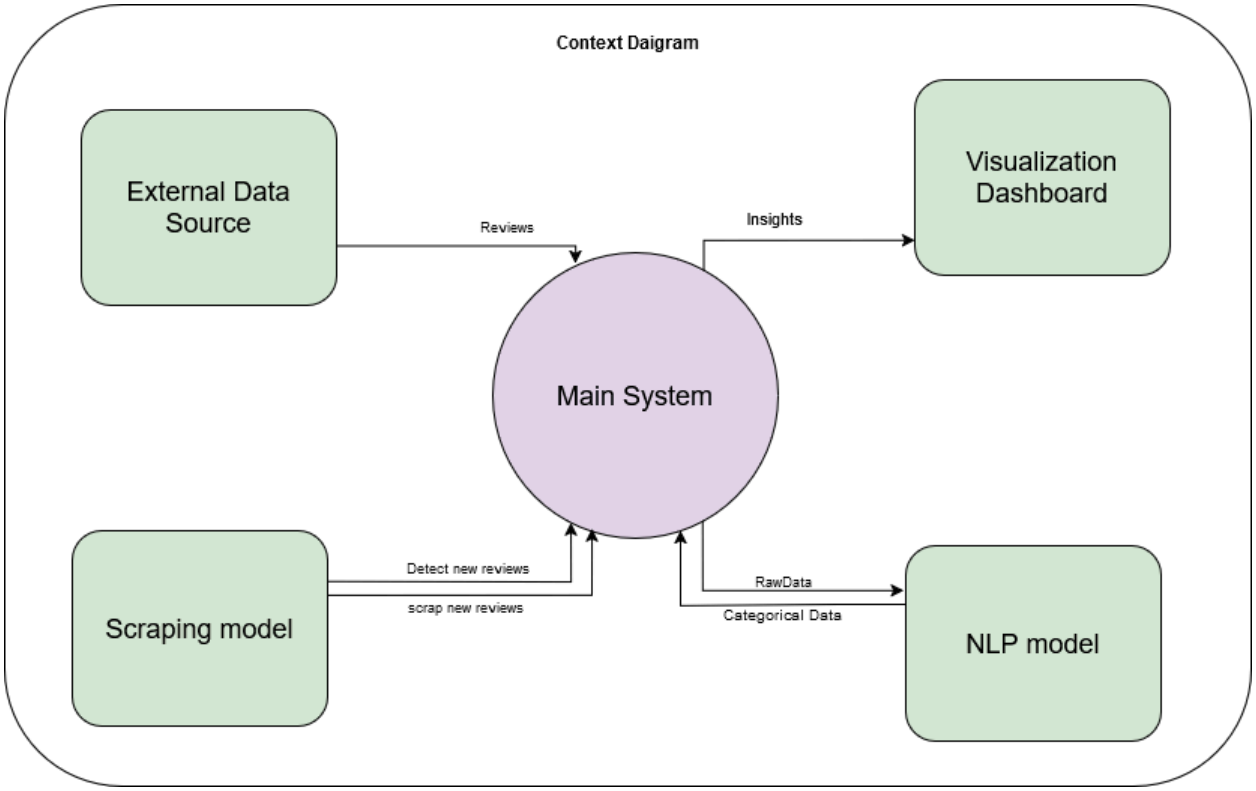


Figure 6 Context

The Smart Tourism Development System is a data-driven platform designed to enhance Egypt's tourism sector by collecting and analyzing tourist feedback from external sources like TripAdvisor. The system uses automated scraping to detect and extract new reviews, processes them with NLP models for sentiment analysis and categorization, and transforms the data into actionable insights. These insights are visualized through interactive dashboards, providing stakeholders (government, tourism operators, and local businesses) with revenue trends, service quality metrics, and areas for improvement. By leveraging cloud infrastructure and machine learning, the system supports data-driven decision-making, improves tourist satisfaction,

and promotes sustainable tourism growth while preserving Egypt’s cultural heritage.

4.4.3: Data flow diagram

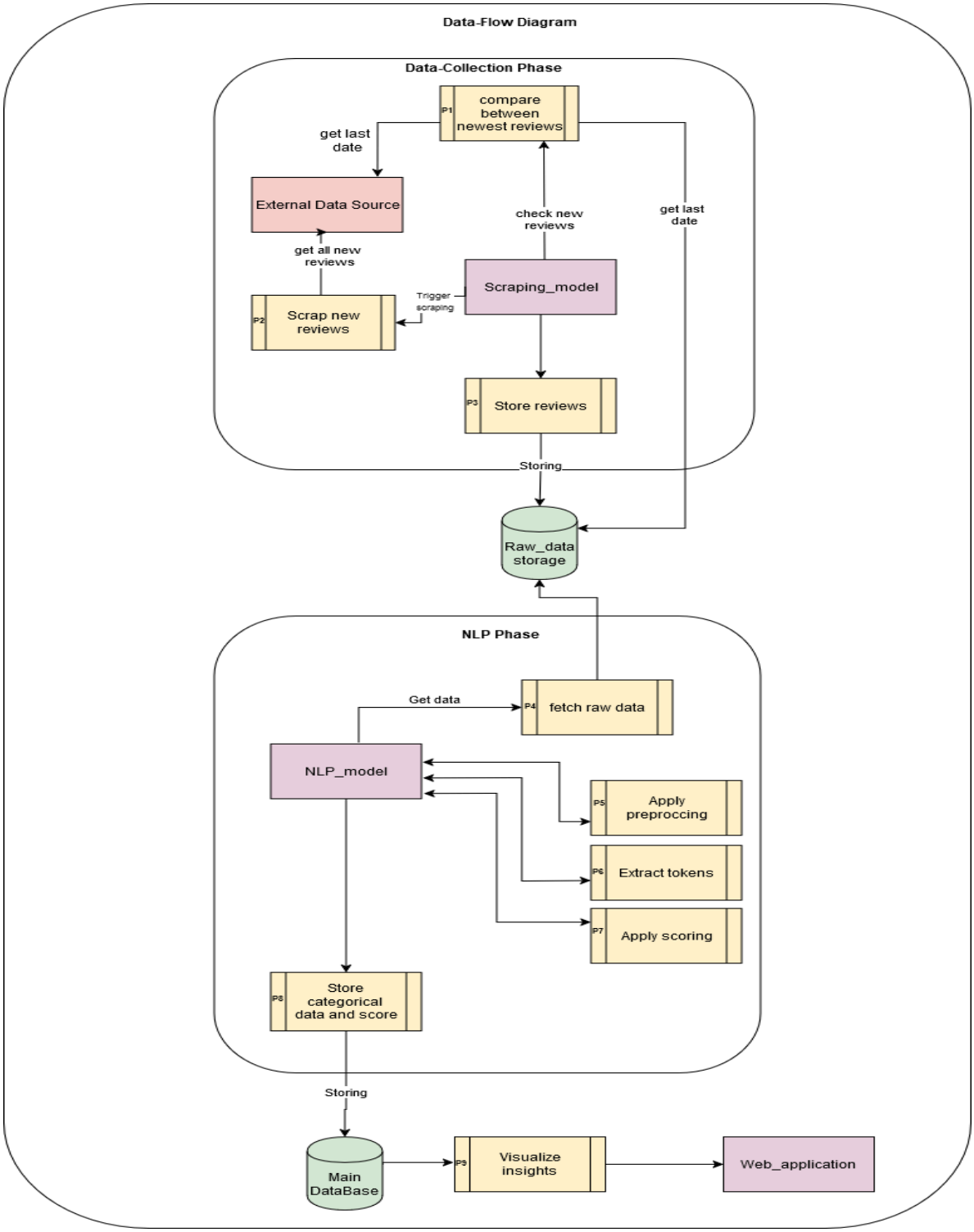


Figure 7: Data flow

1. Data-Collection Phase:
    - The system interacts with an External Data Source to fetch reviews.
    - It compares the last collected review date with new reviews to identify updates.
    - The Scraping Model extracts new reviews and stores them as Raw Data in the database.
2. NLP Phase:
    - Raw data is retrieved and preprocessed (e.g., cleaning, tokenization).
    - The NLP Model applies sentiment scoring and categorizes the data.
    - Processed data (categorical data and scores) is stored back in the database.
3. Visualization Phase:
    - A Web Application retrieves processed data from the database.
    - Insights are visualized through dashboards for stakeholders.
4. Database:
    - Acts as the central storage for raw data, processed data, and categorical insights.

4.5 interaction class diagram

4.5.1 Back-End Sequence Diagram

4.5.1.1 Scraping Sequence Diagram

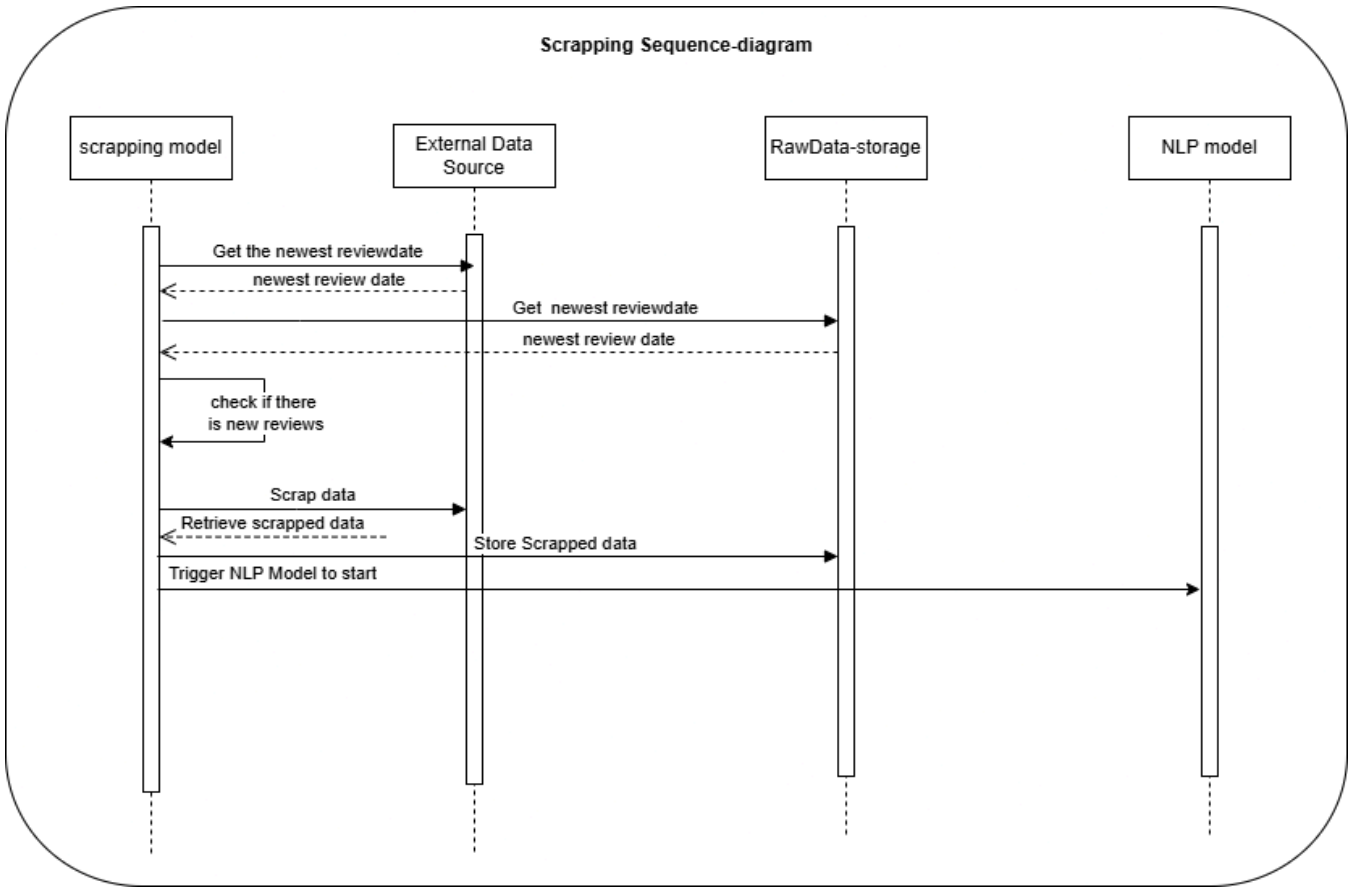
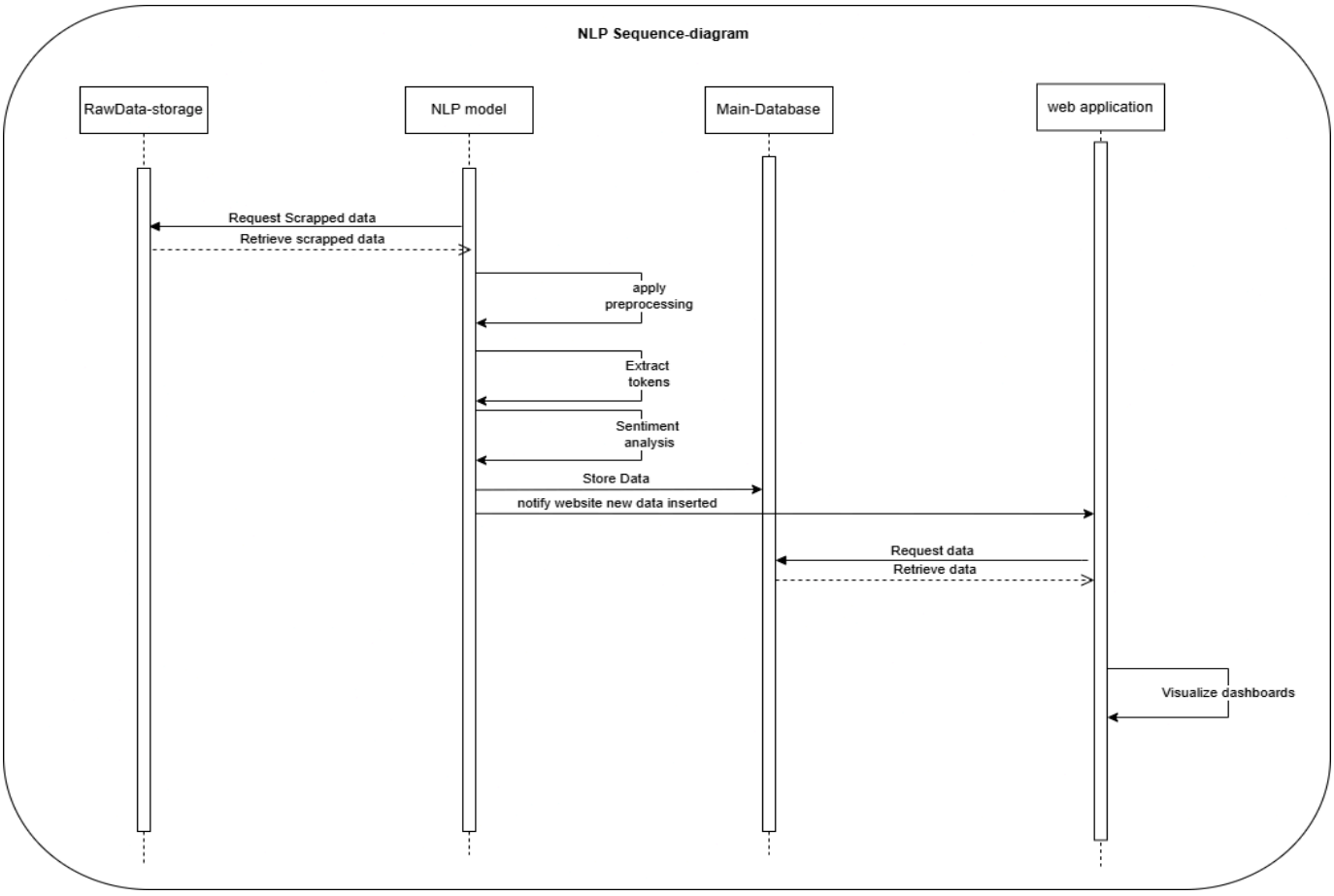


Figure 8 Scraping Sequence Diagram

1. Data Retrieval:
    - The system gets the newest network directory from the Format Data Source.
    - It checks for new data and connects it into a file for processing.
2. Data Scanning and Storage:
    - Scans the data for exceptions and stores the sequence data in the Path Data Storage.

4.5.1.1 Scraping Sequence Diagram



3. NLP Processing:
    - The NLP Model is targeted to process the data:
        - Applies preprocessing (e.g., token extraction).
        - Executes selected scripts for analysis.
4. Data Transfer and Storage:
    - Processed data is sent to the Main Database for storage.
    - The system ensures only necessary data is saved.
5. Web Application Interaction:
    - The Web Application requests data from the database.
    - Retrieves and visualizes insights for stakeholders.

4.5.1 Front-End Sequence Diagram

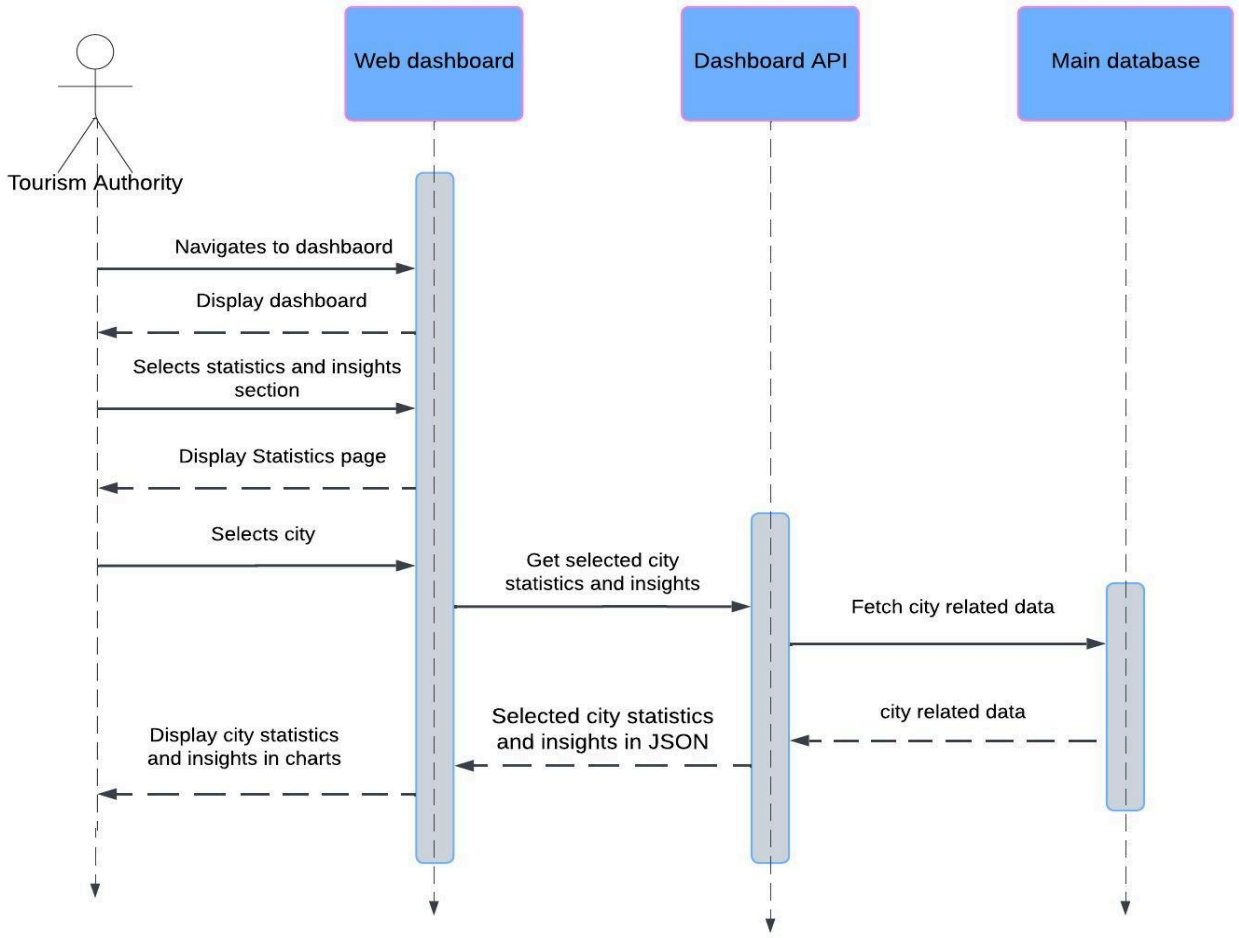


Figure 10 Front-End Sequence

1. **User Interaction**: The Tourism Authority accesses a web dashboard and selects sections such as Statistics and Insights.
2. **Data Request**: Upon selecting a city, the system requests its statistics and insights. The Dashboard API retrieves this data from the Main Database.
3. **Data Processing and Display**: The API processes the city's data, formats it into JSON, and presents it on the dashboard. Insights are visualized using charts for clarity and ease of understanding.

4.6 Design Class

4.6.1 Class Diagram

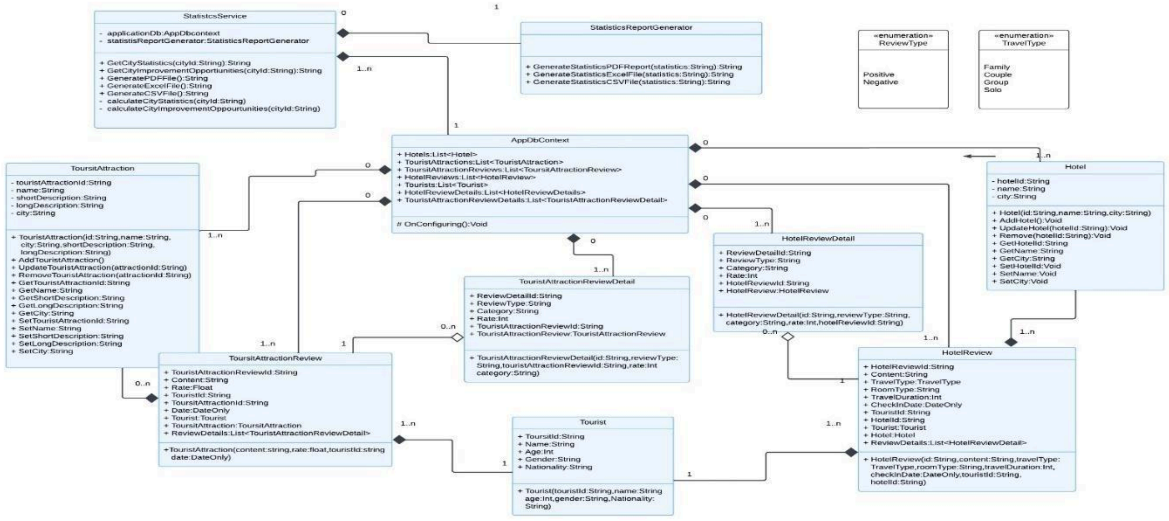


Figure 11 Class

A **Class Diagram** for the Smart Tourism Development System would typically include the following key classes and their relationships:

1. **TouristFeedback**:
    - Attributes: reviewID, reviewText, rating, date, location
    - Methods: getSentiment(), getCategory()
2. **ScrapingModel**:
    - Attributes: sourceURL, lastScrapedDate
    - Methods: scrapeNewReviews(), checkForUpdates()
3. **NLPModel**:
    - Attributes: tokens, sentimentScore, category
    - Methods: preprocessData(), analyzeSentiment(), categorizeFeedback()
4. **Dashboard**:
    - Attributes: userRole, cityFilter, displayMode
    - Methods: displayStatistics(), generateCharts(), filterByCity()
5. **Database**:
    - Attributes: rawData, processedData, insights
    - Methods: storeData(), retrieveData(), updateData()
6. **TourismAuthority**:
    - Attributes: userID, accessLevel
    - Methods: viewDashboard(), generateReports()

4.7 Database Schema

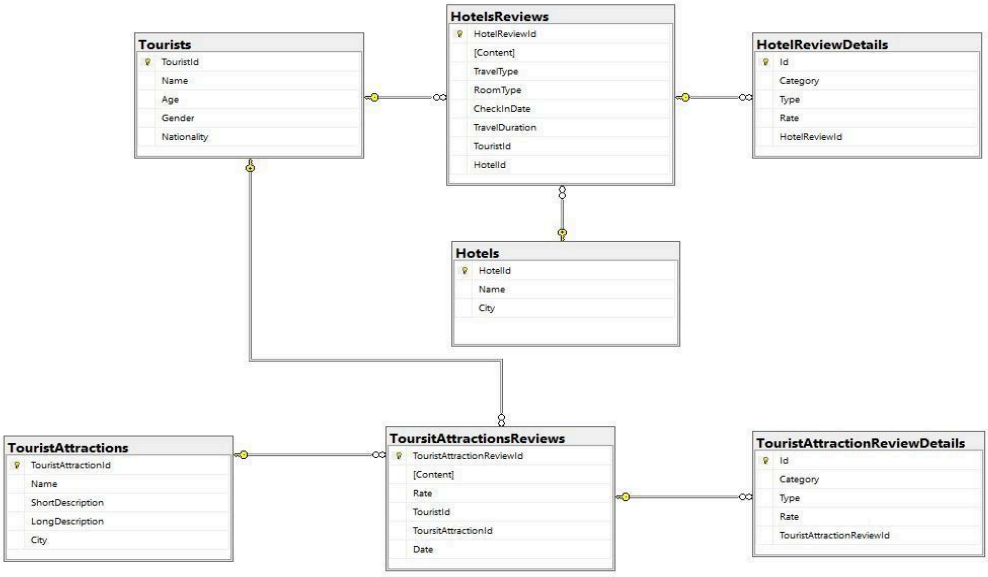


Figure 12 DataBase Schema

The database schema for the system is designed to store and manage data related to tourist feedback, insights, and analytics. Key tables include:

1. TouristFeedback:
    - Columns: ReviewID (Primary Key), ReviewText, Rating, Date, Location, SentimentScore, Category
    - Stores raw and processed feedback from tourists.
2. Cities:
    - Columns: CityID (Primary Key), CityName, Region, TouristCount
    - Contains information about cities and their tourism statistics.
3. Insights:
    - Columns: InsightID (Primary Key), CityID (Foreign Key), RevenueTrend, ServiceQualityScore, FeedbackSummary
    - Stores analyzed insights and trends for each city.
4. Users:
    - Columns: UserID (Primary Key), Username, PasswordHash, Role
    - Manages user accounts for stakeholders (e.g., tourism authorities, operators).
5. DashboardData:
    - Columns: DashboardID (Primary Key), CityID (Foreign Key), VisualizationType, DataJSON

- Stores pre-processed data for visualization on the dashboard.

4.8 ER Diagram

4.8.1 Main_DataBase

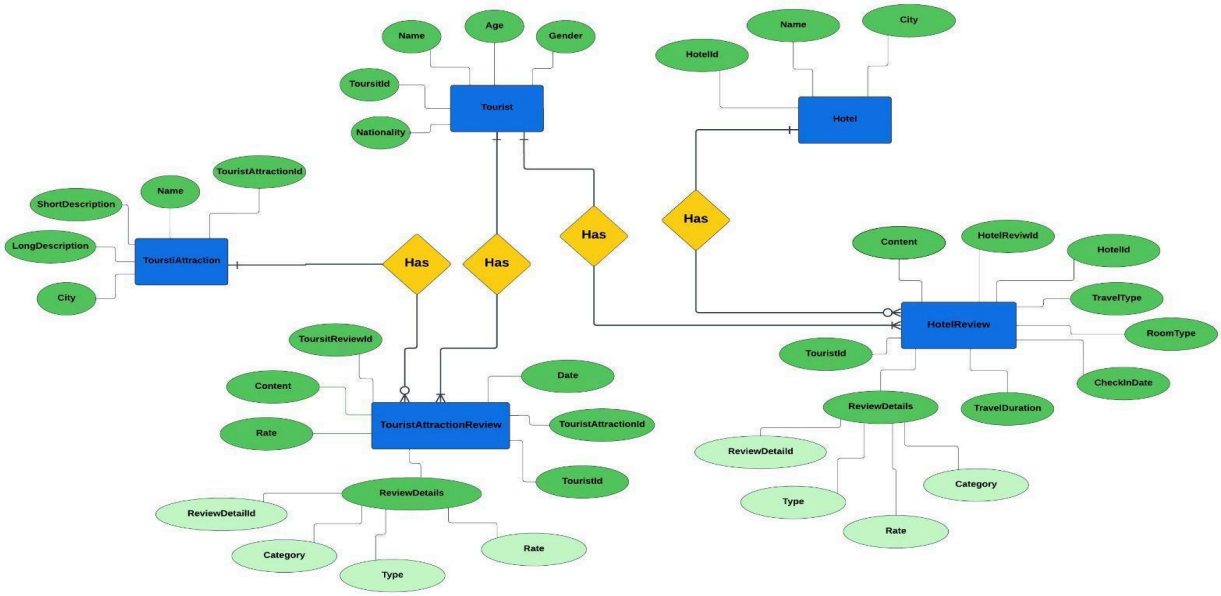


Figure 13 Main DataBase-ERD

The ERD of the Smart Tourism Development System models relationships between Tourists, Hotels, and Tourist Attractions, focusing on collecting and analyzing reviews to improve Egypt's tourism sector. Key entities include Tourist, Hotel, and Tourist Attraction, each with attributes like ID, name, and location details. Tourists can leave reviews for both hotels and attractions, with review details covering aspects such as content, category, and rating.

The system supports functionalities like data collection, tourist behavior analysis, sentiment analysis, and decision-making support to enhance service quality and visitor satisfaction. The structured data model ensures efficient management and actionable insights for stakeholders.

4.7 Mock-UP

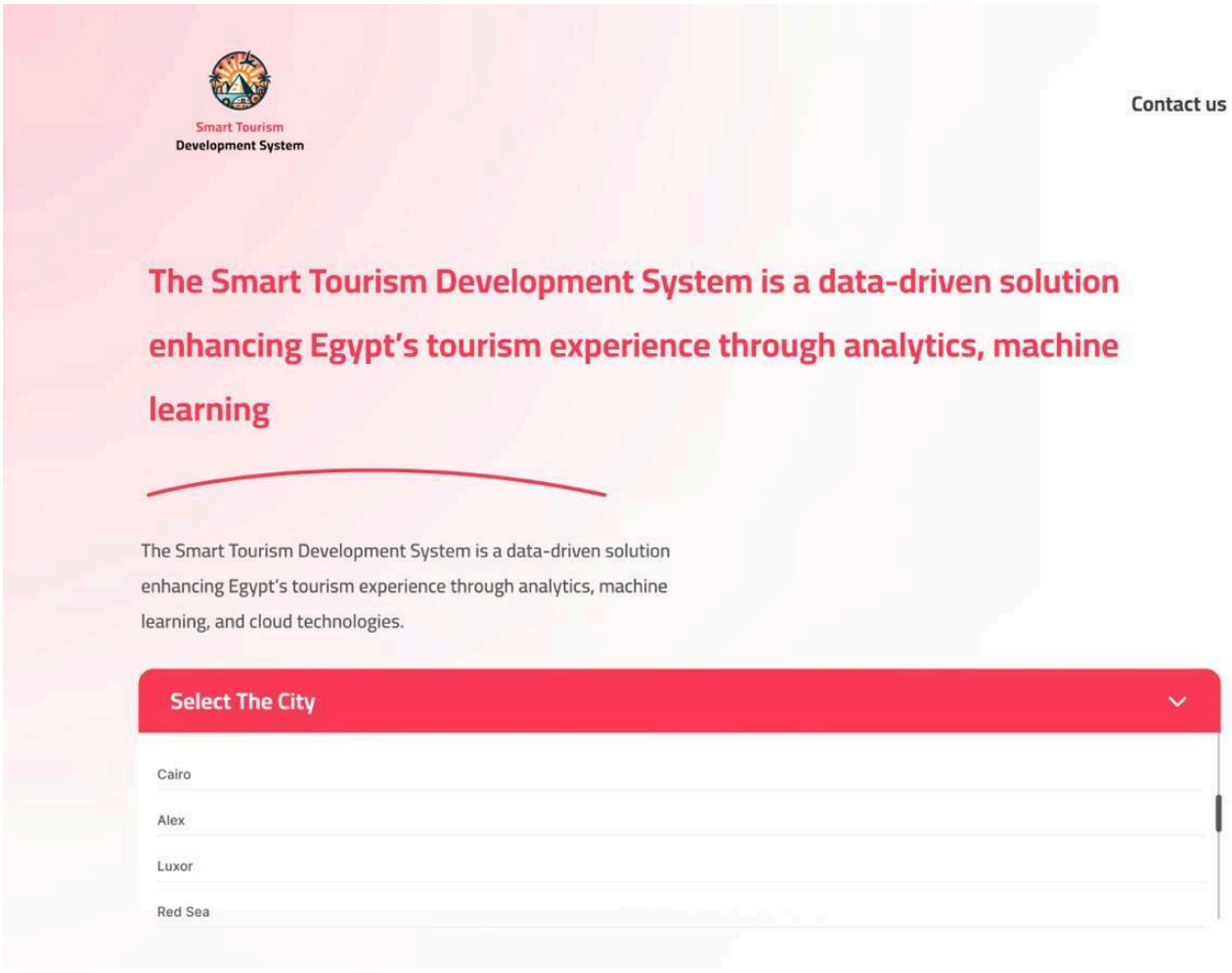
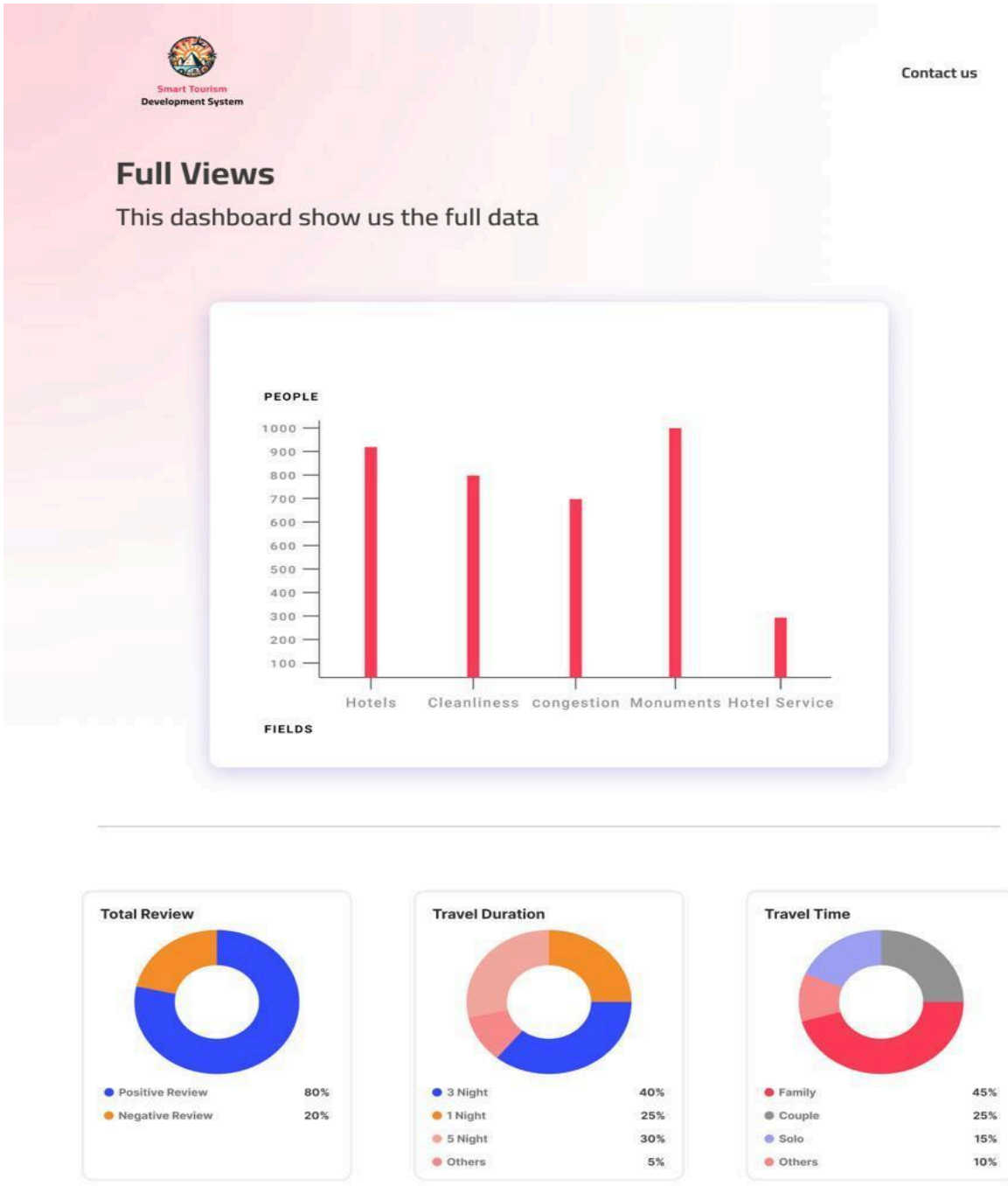


Figure 14 Mock-UP

Page 1: Overview Page

- **Header:**
  The header introduces the Smart Tourism Development System, emphasizing its role as a data-driven solution that enhances Egypt's tourism experience using analytics, machine learning, and cloud technologies.

- **City Selection Dropdown:**
  Users can select a city from the dropdown menu, including Cairo, Alexandria, Luxor, and the Red Sea, to explore tailored insights and data specific to their chosen destination.



This mock-up represents the Smart Tourism Development System’s commitment to leveraging data analytics to enhance Egypt's tourism sector, offering intuitive tools for exploration and actionable insights.

Chapter Conclusion

In this chapter, we have outlined the proposed system's functionality and design, detailing how key operations such as scraping, sentiment analysis, and generating insights will be executed. Through well-defined use cases, we illustrated how the system interacts with data sources and stakeholders. The system architecture and design diagrams provide a clear understanding of the structure, data flow, and inter-component communication. These elements collectively ensure that the Smart Tourism Development System can effectively address the challenges in Egypt's tourism sector, offering actionable insights and data-driven solutions for stakeholders.

Chapter 5

System Implementation

5.1 Overview

This chapter outlines the practical implementation of the Smart Tourism Development System. It details how each system component was developed, integrated, and tested to ensure end-to-end functionality. The implementation spans backend development, data processing workflows, frontend dashboards, automation pipelines, and system deployment.

5.2 Development Environment

Component	Tools & Technologies
Programming Language	Python 3.10
Frontend Framework	Streamlit
Backend & APIs	Streamlit APIs only
Data Processing	Pandas, NLTK, Scikit-learn ,Langdetect, Deep-Translator
Visualization	Plotly, Seaborn, Matplotlib,Dash,Collections, Ast, Pages
Automation	Apache Airflow
Databases	PostgreSQL, (SQL Data Lake)
Version Control	GitHub

5.3 Backend Implementation

The backend handles data ingestion, preprocessing, sentiment analysis, and data storage:

- Web Scraping: Implemented using BeautifulSoup and Selenium to extract reviews from TripAdvisor and Booking.com.

- Preprocessing: Custom scripts clean text, remove stop words, normalize whitespace, and handle mixed language inputs.
- Sentiment Analysis: Using TF-IDF vectorization and rule-based models with NLTK to assign sentiment scores (-1 to 1).

-Database: Implemented using PostgreSQL to be a mediator between Data Preprocessing and Frontend Implementation.

Scripts are modularized as:
- scraper.py: Crawls and stores new reviews.
- preprocess.py: Cleans and transforms review text.
- nlp_model.py: Performs tokenization and scoring.

5.4 Frontend Implementation

The frontend is built using Streamlit, offering a lightweight yet interactive interface:

- City Dashboard: Displays tourist sentiment insights for each city.
- Attraction Dashboard: Shows advantages/disadvantages of tourist spots.
- Filters: Users can select cities, categories, and time periods.
- Dynamic Charts: Created using Plotly and Seaborn.

Each page in Streamlit communicates with the data source and Redis cache to fetch the latest data.

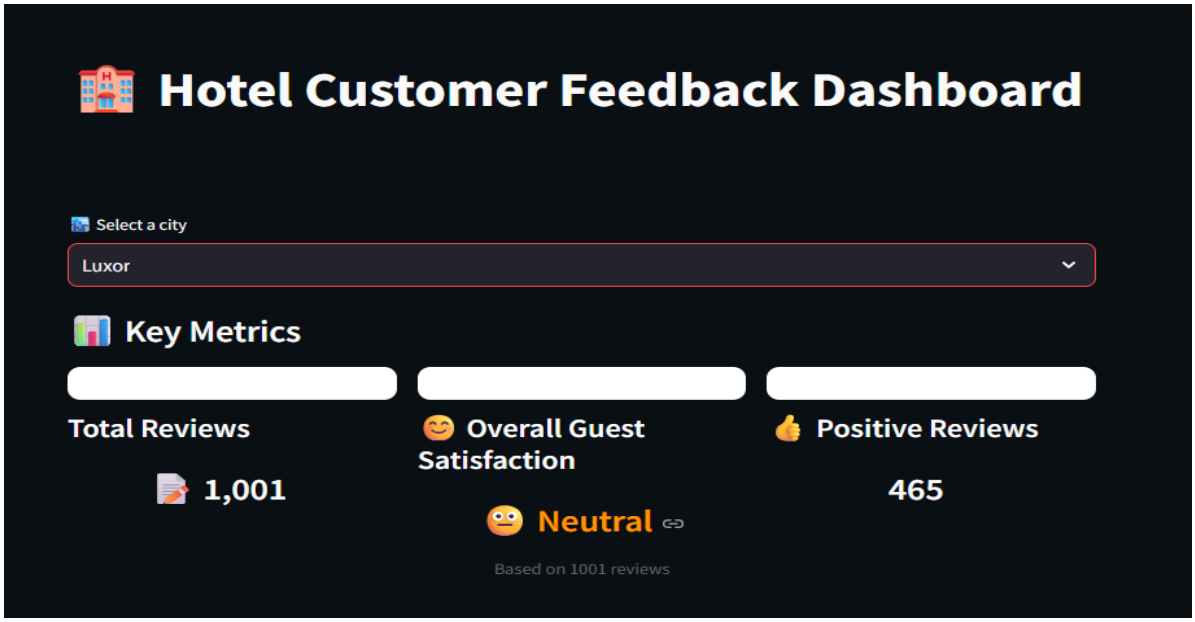
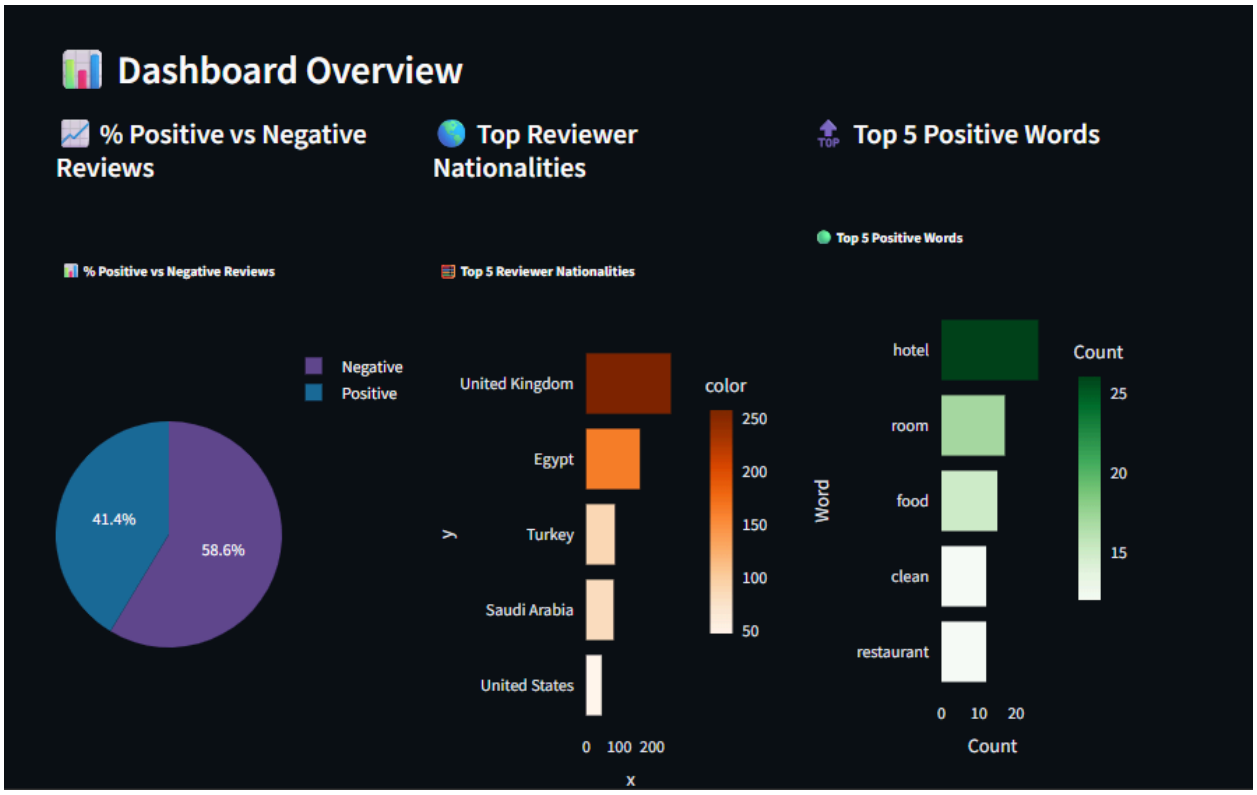

Hotel Customer Feedback Dashboard
Select a city
Luxor
Key Metrics
Total Reviews
1,001
Overall Guest Satisfaction
Neutral
Based on 1001 reviews
Positive Reviews
465

figure 14


Dashboard Overview
% Positive vs Negative Reviews
Top Reviewer Nationalities
Top 5 Positive Words
% Positive vs Negative Reviews
Negative
Positive
41.4%
58.6%
Top 5 Reviewer Nationalities
United Kingdom
Egypt
Turkey
Saudi Arabia
United States
color
250
200
150
100
50
0 100 200
x
y
Top 5 Positive Words
hotel
room
food
clean
restaurant
Count
25
20
15
0 10 20
Count
Word

5.5 Automation & Workflow Management

To maintain a robust and scalable pipeline, Apache Airflow was used:

- DAGs (Workflows):
  - check_new_reviews_dag: Detects new reviews daily.
  - scraping_dag: Triggers scraping scripts if updates are found.
  - preprocessing_dag: Cleans new raw data.
  - nlp_analysis_dag: Applies sentiment analysis.

All DAGs include retry logic and error logging to AWS S3.

5.6 System Integration

System components are loosely coupled for scalability:

- Processed data is served through Streamlit .

- A caching layer using Redis reduces latency and server load.

- PostgreSQL stores processed insights, and raw data is held in an SQL-based data lake.

5.7 Sample Outputs

Examples of system outputs:

- Sentiment Score Example:

```
{
  "review": "The hotel was clean but overpriced",
  "sentiment_score": 0.2,
  "category": "cleanliness"
}
```

- Dashboard View:
  - Pie charts of sentiment distribution.
  - Top complaints per city.
  - Comparison of cities based on review count and score.

Chapter Conclusion

This chapter demonstrated the full implementation process of the Smart Tourism Development System, covering backend modules, data pipelines, frontend dashboards, and deployment strategies. The modular design, combined with automation and cloud deployment, ensures the system is scalable, efficient, and ready for real-world use by tourism stakeholders.

Chapter 6

Testing & Validation

6.1 Overview

This chapter outlines the testing strategy used to ensure the reliability and performance of the Smart Tourism Development System. Testing covered scraping functions, NLP components, automation via Airflow, and the Streamlit-based dashboards to ensure proper flow and presentation of data.

6.2 Testing Objectives

- Validate the accuracy of the sentiment analysis module.
- Ensure scraping scripts fetch and store reviews correctly.
- Verify data flows smoothly between scraping, processing, and visualization.
- Ensure the Streamlit dashboard updates and renders results as expected.
- Confirm automation DAGs execute tasks reliably and recover from failures.

6.3 Testing Methodology

Type	Description
Unit Testing	Tested individual functions (e.g., sentiment scoring, text cleaning).
Integration Testing	Verified flow from scraping → processing → Streamlit rendering.
System Testing	Ran full end-to-end execution, observing outputs on dashboard.
Performance Testing	Assessed speed and memory usage when loading large datasets.
User Acceptance Testing (UAT)	Final demonstration for supervisors and domain users.

7.4 Sample Test Cases

- Sentiment Analysis
  - Input: `"Very nice hotel, but too crowded."`

- Expected Output:
    - Sentiment Score: ~0.1 (neutral)
    - Categories: `["experience", "crowd"]`
- Result: ✅ Passed

🔹 Scraping & Retry Logic

- Scenario: Website blocks request temporarily.
- Expected: Retry mechanism activates, logs error, then resumes.
- Result: ✅ Passed

🔹 Streamlit Dashboard Rendering

- Action: Select city = "Luxor"
- Expected: Dynamic graphs appear, loaded data reflects current sentiment.
- Result: ✅ Passed

6.5 Tools Used

Tool	Purpose
pytest	Unit testing for scripts
Airflow Logs	DAG monitoring and retry checking
Streamlit	Manual testing for UI functionality
Jupyter Notebook	Debugging and testing ML logic

6.6 Error Handling & Logging

- Scraping: Includes try-except with retry and logging in log files.
- Preprocessing: Logs unprocessable reviews for later manual inspection.
- NLP: Logs failures during tokenization or scoring.
- Dashboard: Streamlit shows fallback messages if no data is available.

6.7 Summary of Test Results

Component	Status	Notes
Scraping (TripAdvisor)	✅ Passed	Handles pagination and retry logic
NLP Preprocessing	✅ Passed	Handles noisy text (emoji/mixed langs)
Sentiment Analysis	✅ Passed	Validated against sample reviews
Streamlit Dashboard	✅ Passed	Updates dynamically, graphs load fast
Automation with Airflow	✅ Passed	DAGs run in sequence with logs

6.8 UAT (User Feedback)

- Presented To: Academic supervisors and project evaluators.
- Feedback:
    - Streamlit dashboard is easy to use and informative.
    - Sentiment insights are relevant and clear.
- Suggestions:
    - Add filtering by date or sentiment type.
    - Improve initial load speed with caching (already implemented via Redis).

6.9 Chapter Conclusion

Testing confirmed that the system operates reliably from data collection through to visualization. With all major components validated—including the NLP model, scraping logic, and Streamlit interface—the Smart Tourism Development System is ready for real-world deployment and stakeholder use.

Chapter 7

Conclusion & Feature work

7.1 Feature Work

To ensure the continuous evolution and enhancement of the Smart Tourism Development System, we have outlined detailed areas of future work:

---

1. Integration with Real-Time Data Sources

Objective: Provide tourists with up-to-the-minute recommendations and dynamic itinerary adjustments.

Implementation:

Integrate APIs for live weather forecasts to help tourists plan outdoor activities.

Use traffic monitoring systems to suggest alternate routes or less crowded attractions.

Implement geolocation features to alert users about nearby points of interest or events.

---

2. Expanded Coverage

Objective: Cover more destinations across Egypt, including rural and less-known attractions.

Implementation:

Partner with local tourism boards to gather data about regional attractions.

Incorporate insights about lesser-known landmarks to promote balanced tourism and reduce pressure on popular sites.

Create detailed profiles for each destination, including cultural, historical, and logistical information.

---

3. Personalized Itineraries

Objective: Tailor tourism experiences to individual preferences and needs.
Implementation:
Develop user profiles based on inputs like interests, travel history, and group type (e.g., family, solo).
Utilize collaborative filtering algorithms to suggest activities similar to what the user previously enjoyed.
Allow users to adjust itineraries in real time based on feedback or changing conditions.

---

4. AR and VR Features

Objective: Enable tourists to explore destinations virtually and make informed travel decisions.
Implementation:
Create AR overlays for mobile devices to provide detailed historical or cultural insights at landmarks.
Develop VR experiences for iconic attractions, enabling tourists to "visit" them virtually before their trip.
Partner with museums and cultural sites to create interactive virtual tours.

---

5. Sustainability Insights

Objective: Promote eco-friendly tourism practices and minimize environmental impact.
Implementation:
Highlight green-certified accommodations, transportation options, and activities.
Track and report a tourist’s carbon footprint based on their itinerary and suggest sustainable alternatives.
Partner with local eco-friendly businesses to encourage responsible tourism.

---

6. Sentiment Analysis and Predictive Trends

Objective: Understand and predict tourist needs and satisfaction levels.
Implementation:

Apply NLP techniques to analyze reviews and feedback from platforms like TripAdvisor and social media.

Build predictive models to forecast peak seasons, popular attractions, and emerging trends.

Provide real-time alerts to stakeholders about potential issues, such as overcrowding or negative feedback.

---

7. Collaboration with Local Businesses

Objective: Strengthen partnerships with local businesses to create value for tourists and the economy.

Implementation:

Develop a marketplace where local businesses can showcase services, offer deals, and connect with tourists.

Include features for local guides, artisans, and restaurants to gain visibility among users.

Facilitate seamless booking options for experiences directly through the platform.

---

8. Multilingual Support

Objective: Cater to a global audience by breaking language barriers.

Implementation:

Translate the platform into multiple languages using advanced translation services like GPT or DeepL.

Provide audio guides and navigation support in multiple languages for convenience.

Offer a live chat option with multilingual support agents or AI-powered language assistants.

---

9. Integration with IoT and Smart Devices

Objective: Leverage smart technologies to improve convenience and connectivity.

Implementation:

Develop integrations with smartwatches for travel reminders, notifications, and navigation assistance.

Implement smart kiosks at tourist spots to provide real-time information and guidance.

Use IoT sensors to monitor crowd density at attractions and provide recommendations accordingly.

---

10. Blockchain for Secure Transactions

Objective: Ensure secure and transparent interactions within the platform.

Implementation:

Use blockchain for recording reviews, ensuring authenticity and preventing fake feedback.

Enable secure payment gateways using blockchain technology for transparency in bookings.

Create a decentralized ledger for managing and verifying user data securely.

---

11. Feedback Loop for Policymakers

Objective: Empower decision-makers with actionable insights from tourists.

Implementation:

Design dashboards for policymakers showing aggregated trends, visitor statistics, and feedback.

Create surveys for tourists to provide suggestions, with responses feeding directly into analytics.

Highlight areas requiring infrastructure improvements or service enhancements.

---

12. Mobile Application Development

Objective: Make the platform more accessible and user-friendly on mobile devices.

Implementation:

Develop a mobile app with offline functionality for itinerary access, maps, and guides.

Use geofencing to send push notifications about nearby attractions or deals.

Enable QR code scanning for quick access to information at tourist sites.

---

13. Global Benchmarking

Objective: Position Egypt as a competitive global tourism destination.

Implementation:

Compare key metrics (e.g., visitor satisfaction, infrastructure quality) with leading tourist destinations.

Use benchmarking data to identify gaps and set performance goals for Egypt’s tourism sector.

Share results with stakeholders to align efforts in achieving international standards.

7.2 Conclusion

Conclusion The development of the Smart Tourism Development System marks a significant advancement in the application of modern technology to address longstanding challenges within Egypt's tourism industry. Throughout the project, a comprehensive approach was taken to analyze current limitations and implement effective solutions that harness the power of data analytics, machine learning, and cloud computing. The system provides a structured and scalable framework that allows tourism stakeholders to gain deeper insights into visitor experiences, operational performance, and emerging trends. By automating data collection, processing feedback efficiently, and delivering actionable intelligence, the system enhances the capacity for informed decision-making and strategic planning. One of the key achievements of this project is the ability to bridge the gap between raw data and practical implementation. The integration of predictive analytics and role-based dashboards enables various stakeholders, from government bodies to local businesses, to tailor their strategies based on accurate, real-time information. This not only improves service quality but also contributes to the long-term sustainability of the tourism sector. While the system introduces

numerous benefits, the journey toward optimizing Egypt's tourism landscape remains ongoing.

7.3 References

1. Selenium Documentation Team, "Selenium WebDriver Documentation," SeleniumHQ, 2024. [Online]. Available: https://www.selenium.dev/documentation/

2. Pandas Development Team, "Development — pandas documentation," pandas Documentation, 2024. [Online]. Available: https://pandas.pydata.org/docs/development/index.html#development

3. Python Software Foundation, "re — Regular expression operations," Python Documentation, 2024. [Online]. Available: https://docs.python.org/3/library/re.html

4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., &

5. Duchesnay, E., "Scikit-learn: Machine Learning in Python," scikit-learn Documentation, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E., "TfidfVectorizer: Convert a collection of raw documents to a matrix of TF-IDF features," scikit-learn Documentation, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

7. Bird, S., Klein, E., & Loper, E., "Natural Language Toolkit (NLTK)," nltk Documentation, 2024. [Online]. Available: https://www.nltk.org/

8. MongoDB, Inc., "MongoDB Manual - Tutorials," MongoDB Documentation, 2024. [Online]. Available: https://www.mongodb.com/docs/manual/tutorial/

9. K. Chodorow and M. Dirolf, "MongoDB: The Definitive Guide (3rd ed.)", O'Reilly Media, 2019. [Online]. Available: https://www.oreilly.com/library/view/mongodb-the-definitive/9781491954461/

10. B. Inmon, Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump, Technics Publications, Apr. 1, 2016. [Online]. Available: https://www.amazon.com/Data-Lake-Architecture-Designing-Avoiding/dp/1634621174.

11. Microsoft, "Designing and Implementing Modern Data Architecture on Azure Cloud," Azure Architecture Blog, vol. 2023, no. 9, Sep. 2023. [Online]. Available: https://techcommunity.microsoft.com/blog/azurearchitectureblog/designing-and-implementing-modern-data-architecture-on-azure-cloud-/3440322.

12. G. Patterson, "The Cloud Data Lake: A Guide to Building Robust Cloud Data Architecture," O'Reilly Media, 2023. [Online]. Available: https://books.google.com/books/about/The_Cloud_Data_Lake.html?id=jkuhEAAAQBAJ

13. T. Richards, Streamlit for Data Science: Create Interactive Web Apps for Data Science, June 6, 2023. Available at: https://www.amazon.com/Streamlit-Data-Science-Create-interactive-ebook/dp/B0BTHRBC2W

14. J. Patel and M. Shah, "Developing interactive web applications with Streamlit for data science," J. Data Science Technol., vol. 11, no. 3, pp. 159–172, 2023, doi: 10.1007/s12345-023-00301-3

15. Amazon Web Services, "What is Amazon EC2?" AWS Documentation, 2024. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

16. Amazon Web Services, "What is Amazon Relational Database Service (Amazon RDS)?" AWS Documentation, 2024. [Online]. Available: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html

17. J. Barr and N. Samwal, "Amazon EC2: A Comprehensive Overview," Cloud Computing Advances, vol. 15, no. 3, pp. 45–56, 2023, doi: 10.1007/s10586-023-03210-5

18. L. Wang, J. Tao, and M. Kunze, "Performance Evaluation of Amazon RDS for Large-Scale Applications," Journal of Cloud Computing, vol. 12, no. 2, pp. 123–135, 2024, doi: 10.1186/s13677-024-00234-8

19. L. S. Bobbitt, "UML 2.0 for Beginners", CreateSpace Independent Publishing Platform, Dec. 21, 2021. Available at: https://www.amazon.com/UML-2-0-Beginners-Leonard-Bobbitt/dp/1642959025

20. R. Miles and K. Hamilton, "Learning UML 2.0: A Pragmatic Introduction to UML", O'Reilly Media, Dec. 13, 2006. Available at: https://www.oreilly.com/library/view/learning-uml-20/0596009828/

21. Fuchs, Kevin. “Exploring the opportunities and challenges of NLP models in higher education: is Chat GPT a blessing or a curse?” Frontiers in Education (2023)

22. Mirta, Brítez et al. “The ChatGPT: Revolutionizing Research with AI.” Anais da Academia Brasileira de Ciencias 96 3 (2024): e20230862