Pakistan Food Price Analysis

Design Document

Version 1.0

Group Id: S230286E4C

Supervisor Name: Tayyaba Sehar

Date	Version	Description	Author
(13/09/2023)	1.0	Design Document of Project	(BC190405101,BC190404730)

Table of Contents

Introduction of Design Document

Purpose:

This Design Document serves as a comprehensive guide to the architecture and design of the "Pakistan Food Price Analysis" project. It outlines the structural components, data flow, interaction sequences, class definitions, user interfaces, and testing strategies integral to the successful development of the system.

Scope:

This document encompasses key design aspects that are pivotal to the project's development. It includes Entity Relationship Diagram (ERD), Sequence Diagrams, Architecture Design Diagram, Class Diagram, Interface Design, and Test Cases. Each section provides in-depth insights into the respective design elements.

Intended Audience:

The Design Document is primarily intended for the project development team, including developers, analysts,, supervisors and testers, to ensure a comprehensive understanding of the design and implementation. Additionally, it serves as a reference for stakeholders and researchers interested in the project's technical facets.

Key Design Components:

This document explores various design components critical to the project's success:

1. **Entity Relationship Diagram (ERD):** Illustrates the relationships between entities and data structures within the system, aiding in database design and understanding data flow.

2. **Sequence Diagrams:** Visualize the interactions and message sequences among system components, facilitating a clear understanding of data processing and flow.

3. **Architecture Design Diagram:** Provides a high-level overview of the system's structure and component interactions, helping in system organization and development planning.

4. **Class Diagram:** Offers an in-depth representation of classes and their relationships, aiding in object-oriented design and implementation.

5. **Database Diagram:** The database schema represents the underlying structure for the "Pakistan Food Price Analysis" project. It defines the tables, their relationships, and attributes necessary to store and manage data efficiently.

6. **Interface Design:** Describes user interfaces and visualizations, offering insight into how users interact with the system and how data is presented.

7. **Test Cases**: Outlines the strategies and scenarios for validating and verifying the functionality of the system, ensuring its robustness and reliability.

The subsequent sections of this Design Document delve into each of these design elements, providing detailed insights, diagrams, and explanations to aid in the project's development.

Entity Relationship Diagram (ERD)

Overview:

The Entity Relationship Diagram (ERD) presented in this section provides a visual representation of the data entities, their attributes, and the relationships that govern the "Pakistan Food Price Analysis" project. The ERD is a fundamental component of the system's data structure, illustrating how data entities are interrelated.

Entities:

1. **FoodItem:**

- FoodItemID (Primary Key)
- Name
- Category
- ProductionRegion
- ProductionYear

2. **NonFoodItem:**

- NonFoodItemID (Primary Key)
- Name
- Category
- ProductionRegion
- ProductionYear

3. **Province:**

- ProvinceID (Primary Key)
- Name
- Population

4. **EconomicIndicator:**

- IndicatorID (Primary Key)
- Name
- Description

5. **PriceRecord:**

- RecordID (Primary Key)
- Date
- Price

6. **Associative Table:**

- FoodItemPriceRecord (Associative Table):
- FoodItemPriceRecordID (Primary Key)
- FoodItemID (Foreign Key referencing FoodItem)
- RecordID (Foreign Key referencing PriceRecord)

The data model represents a system for analyzing and recording information related to food items, non-food items, provinces, economic indicators, and their respective price records.

FoodItem and NonFoodItem Entities:

Capture details about different types of items, including their names, categories, production regions, and production years.

Province Entity:

Stores information about provinces, including a unique identifier (ProvinceID), name, and population.

EconomicIndicator Entity:

Represents various economic indicators, each identified by a unique IndicatorID, with a name and description.

PriceRecord Entity:

Records the prices of items at specific dates, with a unique RecordID, date, and price.

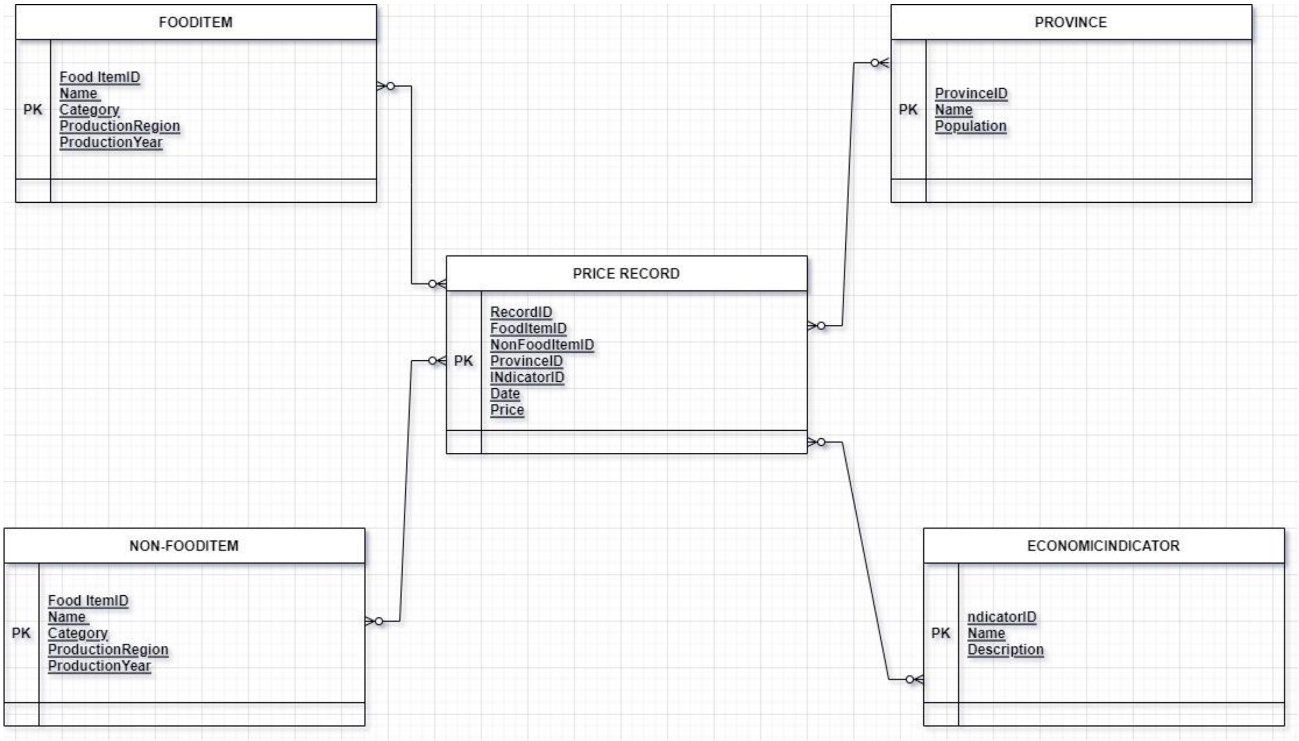
FoodItemPriceRecord Associative Table:

Facilitates a many-to-many relationship between FoodItem and PriceRecord. This table associates specific food items with price records.

Conclusion:

This data model allows for the recording of price information for both food and non-food items over time, associating each price record with the corresponding item. The use of an associative table is essential to handle the many-to-many relationship between FoodItem and PriceRecord. The Province and EconomicIndicator entities provide additional context to the data, allowing for a comprehensive analysis of factors influencing prices.

.


FOODITEM
PK
Food ItemID
Name
Category
ProductionRegion
ProductionYear
PROVINCE
PK
ProvinceID
Name
Population
PRICE RECORD
PK
RecordID
FoodItemID
NonFoodItemID
ProvinceID
INdicatorID
Date
Price
NON-FOODITEM
PK
Food ItemID
Name
Category
ProductionRegion
ProductionYear
ECONOMICINDICATOR
PK
ndicatorID
Name
Description

Sequence Diagram

Overview:

This sequence diagram illustrates the interactions between an actor, represented as "User," and several system components or lifelines within a data analysis system. The diagram captures a series of actions and messages exchanged over time as the user interacts with the system.

Lifelines:

1. **User:** This lifeline represents the external user or actor who initiates interactions with the system. The User performs actions and sends requests to various system components.
2. **Data Collector:** The Data Collector is a system component responsible for retrieving data from external sources or databases. It listens for requests from the User and responds by retrieving the requested data.
3. **Data Preprocessor:** The Data Preprocessor is another system component that validates and cleanses the raw data received from the Data Collector. It ensures that the data is accurate and suitable for analysis.
4. **Data Analyzer:** The Data Analyzer is responsible for applying data mining algorithms to the preprocessed data. It analyzes the data to uncover patterns, trends, or insights based on user requests.
5. **Visualization Module:** The Visualization Module is a component responsible for generating visualizations based on the analysis results. It creates charts, graphs, and other visual representations of data for user comprehension.
6. **Documentation Module:** The Documentation Module is responsible for creating project documentation, including reports, summaries, and analysis documentation. It generates and delivers documentation based on user requests.

Sequence of Interactions:

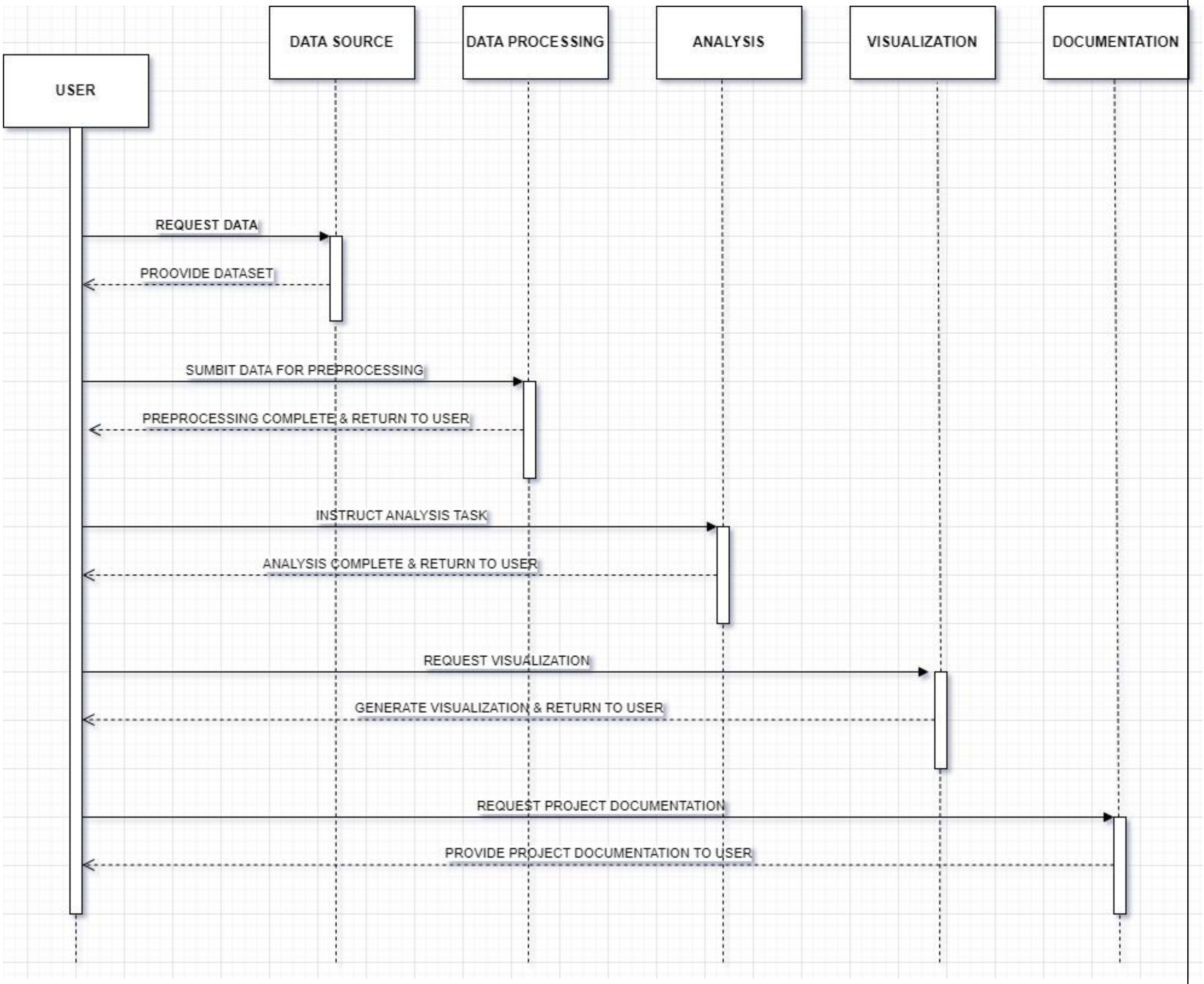
1. The sequence begins with the User sending a request for data analysis, indicating the specific data of interest.
2. The User initiates the interaction by sending a message to the Data Collector, requesting data retrieval.
3. Upon receiving the User's request, the Data Collector retrieves the data from external sources or databases and sends it back to the User.
4. The User validates and cleanses the received data by interacting with the Data Preprocessor, which ensures data quality.
5. The User then instructs the Data Analyzer to apply data mining algorithms to the preprocessed data.
6. The Data Analyzer performs the data analysis and generates insights based on the user's request.

7. The User requests visualizations of the analysis results and communicates with the Visualization Module.
8. The Visualization Module generates charts and graphs and provides them to the User for visualization.
9. Finally, the User requests project documentation from the Documentation Module, which generates and delivers the required documentation.

Conclusion:

This sequence diagram provides a visual representation of the interactions between the User and various system components, illustrating the flow of messages and actions as the User performs tasks within the data analysis system.

Graphically Representation of Sequence Diagram


DATA SOURCE
DATA PROCESSING
ANALYSIS
VISUALIZATION
DOCUMENTATION
USER
REQUEST DATA
PROOVIDE DATASET
SUMBIT DATA FOR PREPROCESSING
PREPROCESSING COMPLETE & RETURN TO USER
INSTRUCT ANALYSIS TASK
ANALYSIS COMPLETE & RETURN TO USER
REQUEST VISUALIZATION
GENERATE VISUALIZATION & RETURN TO USER
REQUEST PROJECT DOCUMENTATION
PROVIDE PROJECT DOCUMENTATION TO USER

Architecture Design Diagram

Overview:

The Architecture Design Diagram presented here adopts a layered approach to illustrate the high-level structure and flow of the "Pakistan Food Price Analysis" project. This layered architecture encompasses the various components and their interactions, demonstrating how data flows from its source to the final visualization.

Layers:

1. **Data Source (World Food Programme Data Cylinder)**:
    - At the core of the architecture is the data source, represented by the "World Food Programme Data Cylinder." This entity symbolizes the data obtained from the World Food Programme Price Database, which serves as the foundation for the analysis.
2. **Data Retrieval and Preprocessing Layer**:
    - The first layer, depicted on top of the data source, consists of two primary components: Data Retrieval and Data Preprocessing.
    - **Data Retrieval**: This component handles the retrieval of data from the World Food Programme Data Cylinder. It initiates the process by fetching the required data sets.
    - **Data Preprocessing**: Once data is retrieved, it is preprocessed to ensure its quality and suitability for analysis. Preprocessing includes tasks like cleaning, validation, and transformation.
3. **Data Mining and Analysis Layer**:
    - The layer above Data Retrieval and Preprocessing represents Data Mining and Analysis.
    - **Data Mining**: This component leverages advanced algorithms to extract patterns, correlations, and insights from the preprocessed data.
    - **Data Analysis**: The results from Data Mining are analyzed to uncover trends and relationships within the data. This analysis is a critical step in understanding food price dynamics.
4. **Required Analysis and Input Actor**:
    - The actor depicted on the right side of the diagram represents an external entity (e.g., a user or stakeholder) who provides input for the specific analysis required. This input guides the system in tailoring the analysis to meet specific objectives.

5. **Data Visualization Layer**:
    - The topmost layer, adjacent to Data Analysis, represents Data Visualization.
    - **Data Visualization**: This component transforms the analyzed data into meaningful visualizations, such as charts, graphs, and maps. These visualizations provide a clear and understandable representation of the analysis results.

Flow of Data and Interactions:

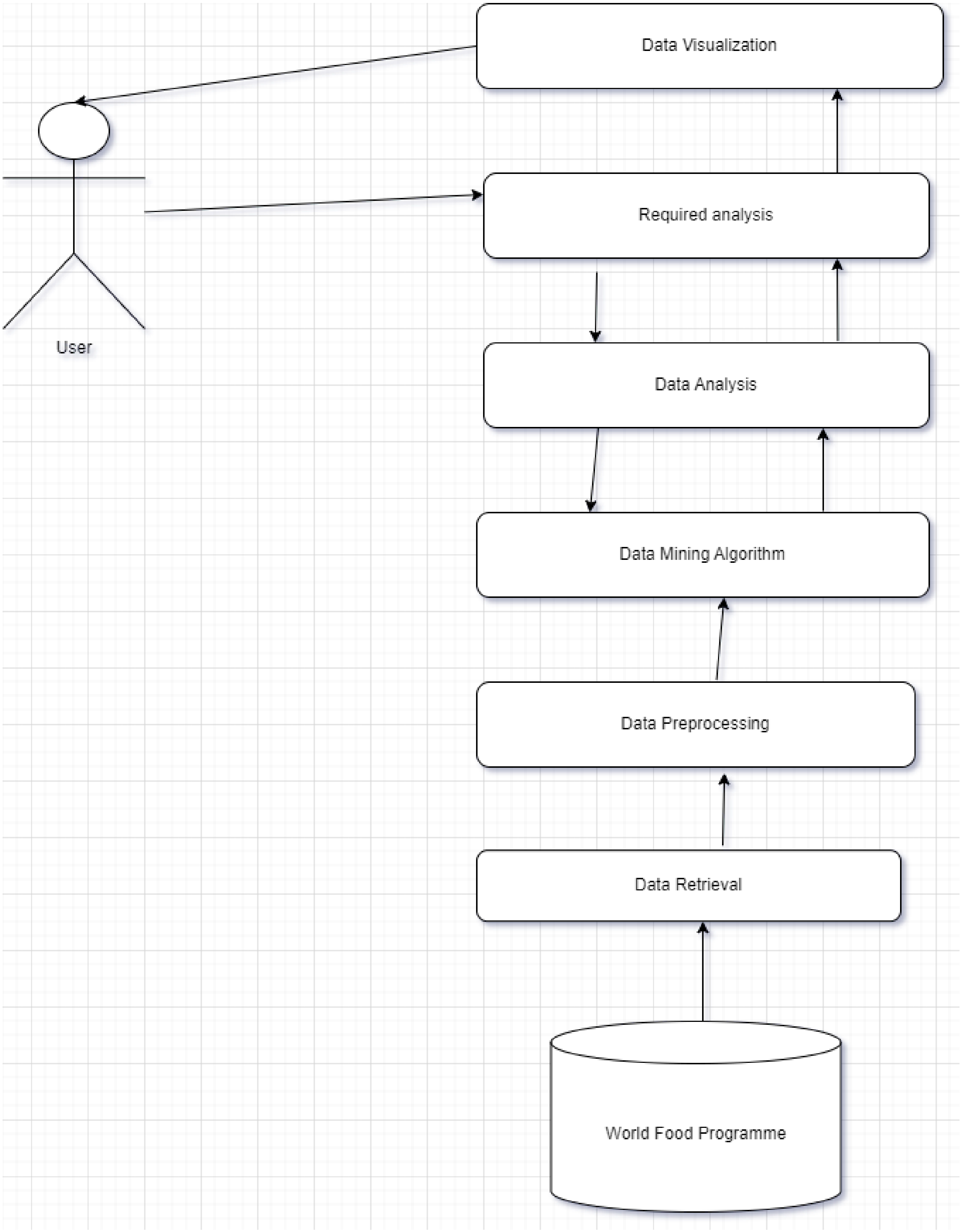
- Data flows from the World Food Programme Data Cylinder through the layers, starting with Data Retrieval and Preprocessing, then moving to Data Mining and Analysis.
- The External Actor provides input for the Required Analysis, which guides the analysis process.
- Once the analysis is complete, the results are visualized and presented to the External Actor.

Notes:

- This layered architecture ensures a well-structured and organized approach to data analysis, starting from data acquisition and preprocessing to advanced analysis and visualization.

Conclusion:

The Architecture Design Diagram with a layered approach provides an overview of the system's structure and data flow in the "Pakistan Food Price Analysis" project. It highlights the key components, their relationships, and the flow of data from its source to meaningful visualizations, facilitating a clear understanding of the project's architecture.


Data Visualization
User
Required analysis
Data Analysis
Data Mining Algorithm
Data Preprocessing
Data Retrieval
World Food Programme

Class Diagram

Overview:

The class diagram illustrates the structure and interactions among various system components or classes within a data analysis system. The diagram provides a visual representation of how these components are organized and how they collaborate to perform tasks.

Classes:

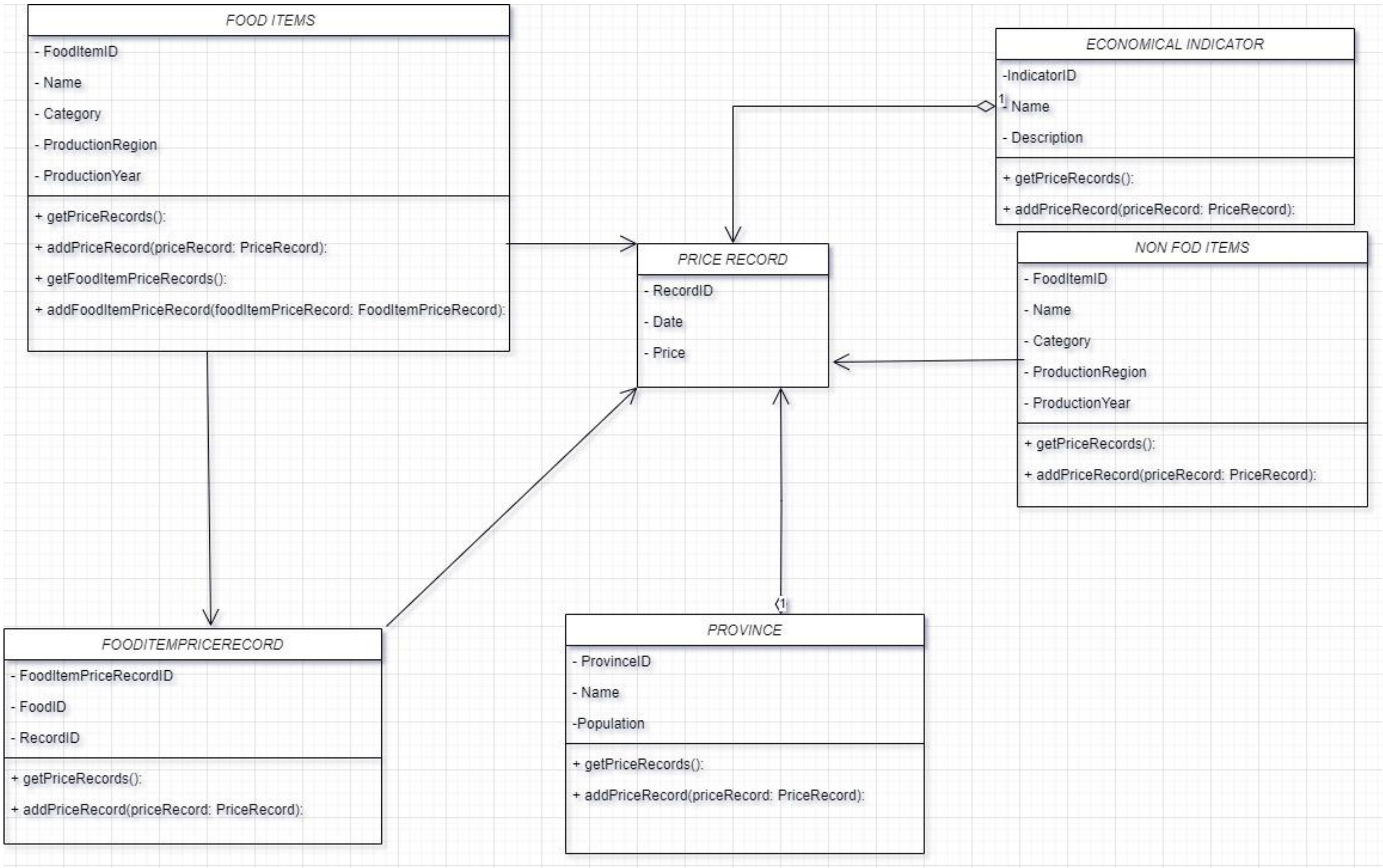
1. **User:** The "User" class represents external users or actors who interact with the system. Users have attributes such as "username" and "password" for authentication. They can perform actions like logging in, requesting data analysis, and viewing results.
2. **DataCollector:** The "DataCollector" class represents a system component responsible for retrieving data from external sources or databases. It contains a method called "retrieveData" to perform data retrieval.
3. **DataPreprocessor:** The "DataPreprocessor" class is another system component. It specializes in validating and cleaning data to ensure its quality. It offers methods like "validateData" and "cleanData."
4. **DataAnalyzer:** The "DataAnalyzer" class represents the component responsible for applying data mining algorithms to the preprocessed data. It contains a method called "applyDataMining."
5. **VisualizationModule:** The "VisualizationModule" class is responsible for generating visualizations based on analysis results. It provides a method called "generateVisualizations" for this purpose.
6. **DocumentationModule:** The "DocumentationModule" class handles the creation of project documentation, including reports and summaries. It offers a method called "createDocumentation."
7. **Database:** The "Database" class represents the data storage and retrieval component of the system. It is associated with various system components, indicating that these components use the database for their operations.

Associations:

- The associations between classes (represented by connecting lines) show how these classes collaborate and interact within the system.
- For example, "User" is associated with "DataCollector," "DataPreprocessor," "DataAnalyzer," "VisualizationModule," and "DocumentationModule" through "Uses" relationships, indicating that the "User" class uses these components to perform its tasks.
- Additionally, all the system components, including "DataCollector," "DataPreprocessor," "DataAnalyzer," "VisualizationModule," and "DocumentationModule," are associated with the "Database" class, indicating that they all use the database for their respective operations.

This class diagram provides an overview of the system's structure, showing how different components interact and collaborate to facilitate user interactions and data analysis processes within the data analysis system.

Graphically Representation of Class Diagram:



DATABASE DIAGRAM

OverView:

The database schema represents the underlying structure for the "Pakistan Food Price Analysis" project. It defines the tables, their relationships, and attributes necessary to store and manage data efficiently. This schema is designed to accommodate data collection, preprocessing, analysis, and reporting aspects of the project.

Tables:

1. **Users Table:**
   - **user_id** (Primary Key): A unique identifier for each user.
   - **username** (Unique): The username used for authentication.
   - **password**: The hashed password for user authentication.
   - Additional user-related attributes (e.g., name, email) can be included.
2. **Data Collection Table:**
   - **data_id** (Primary Key): A unique identifier for each data collection operation.
   - **user_id** (Foreign Key to Users): Links each data collection to a specific user.
   - **data_source**: Describes the source of data (e.g., URL, API).
   - **timestamp**: Records the date and time of data retrieval.
   - Other attributes related to data collection can be included (e.g., location).
3. **Raw Data Table:**
   - **data_id** (Foreign Key to Data Collection): Links raw data to its data collection operation.
   - **raw_data**: Stores the unprocessed, raw data as text or BLOB (Binary Large Object).
4. **Preprocessed Data Table:**
   - **data_id** (Foreign Key to Data Collection): Associates preprocessed data with its data collection operation.
   - **preprocessed_data**: Stores cleaned and validated data as text or BLOB.
5. **Analysis Results Table:**
   - **analysis_id** (Primary Key): A unique identifier for each analysis result.
   - **data_id** (Foreign Key to Data Collection): Connects analysis results to specific data collections.
   - **analysis_type**: Describes the type of analysis (e.g., Time Series, Association Rule Mining).

- **results**: Contains analysis results as text or BLOB.

6. **Visualizations Table:**
    - **visualization_id** (Primary Key): A unique identifier for each visualization.
    - **analysis_id** (Foreign Key to Analysis Results): Associates visualizations with specific analysis results.
    - **visualization_type**: Specifies the type of visualization (e.g., Chart, Graph).
    - **visualization_data**: Stores visualization data as text or BLOB.
7. **Documentation Table:**
    - **documentation_id** (Primary Key): A unique identifier for each documentation entry.
    - **analysis_id** (Foreign Key to Analysis Results): Links documentation to related analysis results.
    - **document_type**: Identifies the type of documentation (e.g., Report, Summary).
    - **document_content**: Contains documentation content as text or BLOB.

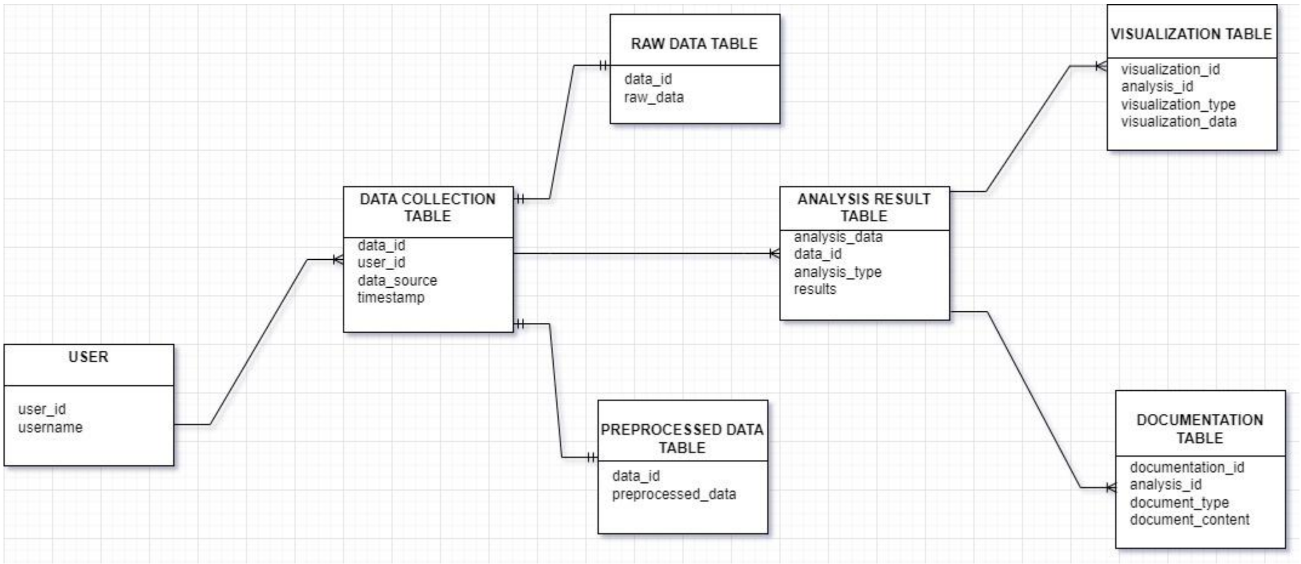
Relationships:

- Users can perform multiple data collection operations, establishing a one-to-many relationship between the **Users** table and the **Data Collection** table.
- Each data collection operation is associated with one set of raw data and preprocessed data, forming one-to-one relationships with the **Raw Data** and **Preprocessed Data** tables.
- Multiple analysis results can be linked to a single data collection operation, creating a one-to-many relationship between the **Data Collection** and **Analysis Results** tables.
- Both the **Visualizations** and **Documentation** tables have one-to-many relationships with the **Analysis Results** table, enabling multiple visualizations and documentation entries for a single analysis result.

Conclusion:

This database schema serves as the foundation for storing, organizing, and retrieving data related to food price analysis in Pakistan. It ensures data integrity and supports various stages of data processing, analysis, and reporting within the project.

Graphical Representation oF Database Diagram



Interface Design

Overview:

The interface design illustrates the graphics with a data visualization system that allows the creation of different types of plots, including pie charts, bar plots, and Seaborn plots. The diagram provides a visual representation of the available options and the flow of interactions within the system.

Components:

1. **Data Visualization Interface:** The "Data Visualization Interface" is the user interface through which users can interact with the system to create data visualizations.
2. **Pie Chart Generator:** This component specializes in generating pie charts based on user input and data sources.
3. **Bar Plot Generator:** The "Bar Plot Generator" is responsible for creating bar plots and bar charts in response to user requests.
4. **Seaborn Plot Generator:** The "Seaborn Plot Generator" represents a module or component that utilizes the Seaborn library to produce sophisticated data visualizations.

Sequence of Interactions:

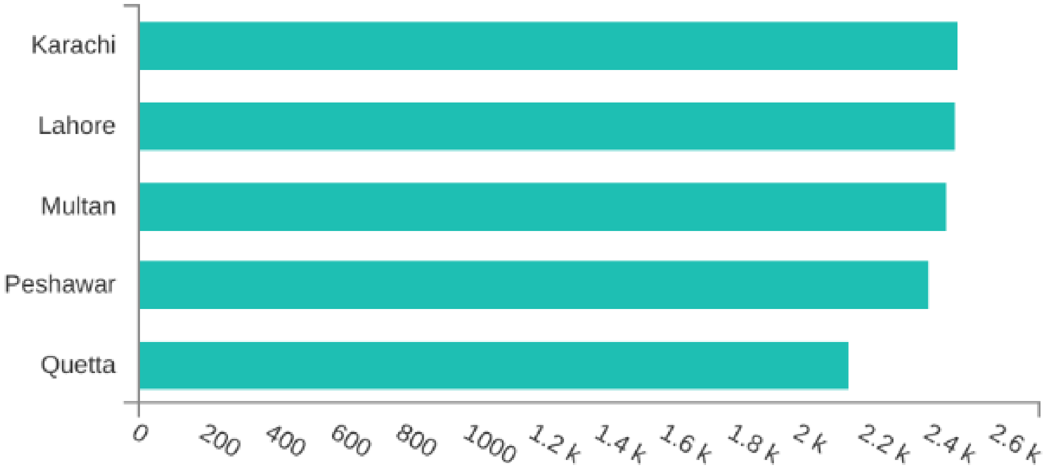
1. Depending on the User's selection, the Data Visualization Interface communicates with the respective plot generator:
   - If a "Pie Chart" is chosen, the Data Visualization Interface sends a request to the "Pie Chart Generator."
   - If a "Bar Plot" is selected, the interface communicates with the "Bar Plot Generator."
   - For "Seaborn Plot," the interface interacts with the "Seaborn Plot Generator."
2. The respective plot generator receives the request, retrieves the necessary data, and generates the specified plot type.
3. The generated plot is then returned to the Data Visualization Interface.
4. Finally, the Data Visualization Interface displays the plot to the User for analysis and interpretation.

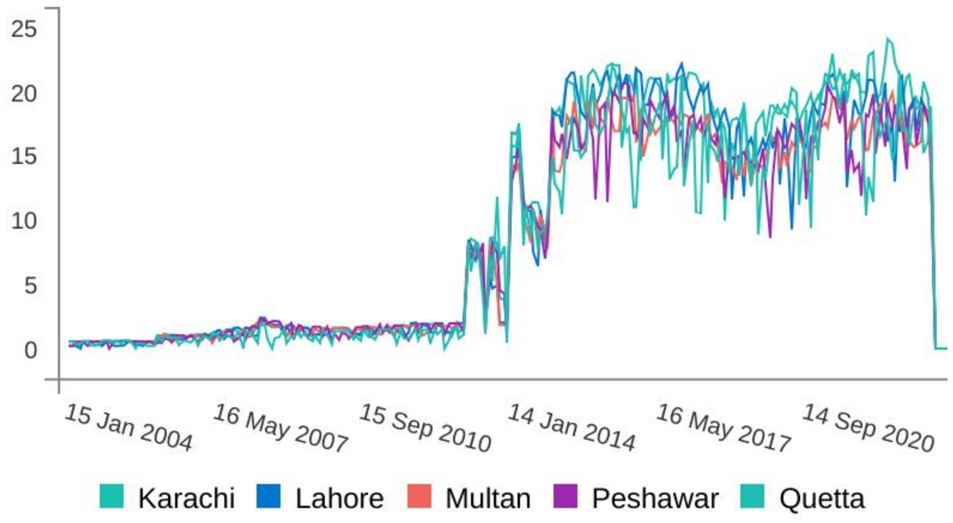
Conclusion:

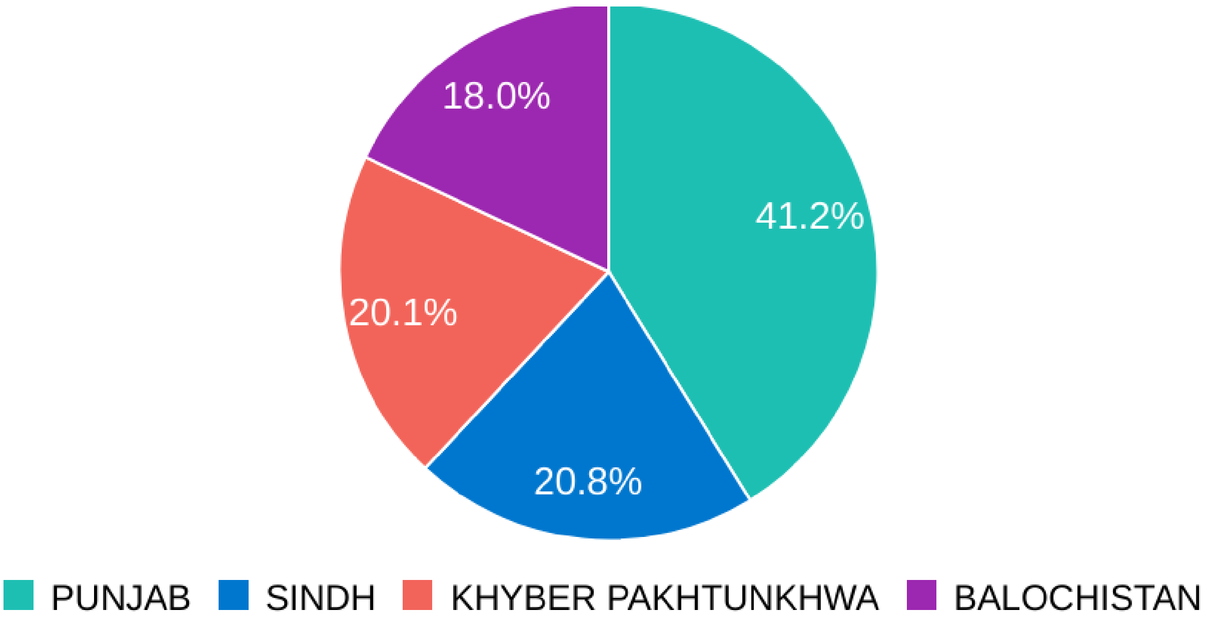
This interface diagram provides an overview of the components involved in creating and presenting various data visualizations to users. It illustrates the flow of interactions, starting from user input to the generation and display of the requested plots, helping users gain insights from their data

Visualization of Data

1. *BAR PLOT*
2. *SEABORN*
3. *PIE CHART*







<u>Test Cases</u>

Test Case 1: Trend Analysis

- *Input*: Historical food price data for different commodities over the past five years.
- *Procedure*: Run the trend analysis algorithm on the input data.

- *Expected Outcome*: The system correctly identifies and visualizes the general trend in food prices over time.

Test Case 2: Relationship Analysis

- *Input*: Sample data for food prices of wheat, rice, and maize.
- *Procedure*: Execute the relationship analysis algorithm to identify correlations.
- *Expected Outcome*: The system successfully identifies and displays significant relationships between wheat, rice, and maize prices.

Test Case 3: Impact of Oil Price Increase

- *Input*: Historical data for food prices and oil prices for the last decade.
- *Procedure*: Run the impact analysis algorithm to quantify the influence of oil price fluctuations on food prices.
- *Expected Outcome*: The system accurately calculates and presents the impact of increased oil prices on various food items.

Test Case 4: Provincial Comparison

- *Input*: Food price data for wheat, rice, and fish across multiple provinces for the past five years.
- *Procedure*: Perform a provincial comparison analysis.
- *Expected Outcome*: The system successfully compares and visualizes price variations among provinces for the selected food items.

Test Case 5: Usability

- *Input*: A non-technical user with minimal guidance.
- *Procedure*: Ask the user to initiate a basic analysis task.
- *Expected Outcome*: The user can easily navigate and perform a simple analysis using the system.

Test Case 6: Performance

- *Input*: A large dataset containing food price data for the past 20 years.
- *Procedure*: Measure the time taken to complete a data mining and analysis task.
- *Expected Outcome*: The system completes the task within a reasonable time frame without significant delays.

Test Case 7: Accuracy and Reliability

- *Input*: Data with known inconsistencies and errors.
- *Procedure*: Assess the system's data preprocessing and cleaning steps.

- *Expected Outcome*: The system effectively identifies and rectifies errors, providing accurate and reliable analysis results.

Test Case 8: Security

- *Input*: An attempt to access data without proper authorization.
- *Procedure*: Test unauthorized access to data.
- *Expected Outcome*: The system denies access and maintains data security as per requirements.

Test Case 9: Scalability

- *Input*: Increase the data volume by adding 10 years of additional historical data.
- *Procedure*: Measure the system's performance and resource utilization.
- *Expected Outcome*: The system scales gracefully, accommodating the larger dataset without performance degradation.