

System Model and Design: Tripedia

Yulin Zhang

7th Group

Software Engineering

Montreal, Canada

silveralex2023820@gmail.com

Yuhang Chen

7th Group

Software Engineering

Montreal, Canada

yuhang.chen@mail.concordia.ca

Jiaxi Yang

7th Group

Software Engineering

Montreal, Canada

yjxyang2@outlook.com

Boyang Wang

7th Group

Software Engineering

Montreal, Canada

wangboyang0626@outlook.com

I. THE AGILE PROCESS

A. Team Setup

B. Association between User Stories and Sub-requirements

C. The Scrum Sprint Cycle

- 1) *Review Work to be done:*
- 2) *Product Backlog:*
- 3) *Sprint Plan:*
- 4) *Sprint Backlog:*
- 5) *Sprint Process:*
- 6) *Sprint Review:*

II. ARCHITECTURE

A. Architecture Design in Overall Level

As shown in Figure1, the ICDE system architecture mainly contains two parts: client and server. Based on the limited server hardware equipment, we implement the data analysis logic as a single module in the server, which stands for ICDE 3rd Party Applications.

In the client part, we implement Tripedia's client logic based on the web. The web client service is mainly divided into two parts: the Data Capture Module and User Assistant Module. The Data Capture Module captures the user's behavior and personal information through questionnaires and behavior captures, and transmits the relevant information to the server. The User Assistant Module is responsible for displaying processed data sent by the server, such as recommendation results and data analysis results, which help users make decisions easily.

On the server side, we will develop three modules to implement the backbone of the ICDE system: Data-processing Module, Data-analyzing Module, and Data Access Service. The Data Access Service, as the median layer for database access, provides a unified database interface (deletion, modification, query, etc) for the Data-processing Module and the Data-analyzing module. The Data-processing Module is responsible for the basic logic of the server, such as registration, login, search, data preprocessing, and other interaction logic with the client. The Data-analyzing Module accesses the data in the database through the Data Access Service to perform data analysis and returns the analysis results to the Data-processing Module.

To sum up, Tripedia's architecture is a standard ICDE system. The Client part includes a Data Capture Service to collect user behavior and information and transmit relevant

data to the server. The Data-processing Module in the Servers section is responsible for implementing the interactive logic of the server; the Data Access Service provides a unified data access interface; the Data-analyzing Module serves as the core of the ICDE system to analyze user data and feedback the analysis results to the user through the server to provide User Assistance Service.

B. Architecture Design in User Requirements Level

In this section, we will discuss the design of the tasks of the Tripedia. For a clearer structural presentation, we consider the choice of task description granularity. A small task granularity can not take on the task of showing interactions among multiple objects. A large task granularity may lose the meaning of the task design. Therefore, we merged some sub-requirements into a single task. We try to show the relationships among objects as much as possible while ensuring appropriate granularity.

III. MODELING AND DESIGN

A. Task: Searching Data from Database

1) Task Context :

The class diagram in Figure 2 shows the relationships among the searcher and database manager.

The DBManager is a base class in the database interface, which encapsulates the Mongo Database interface. The DBManager can search hotel data and ticket data from the database.

The Searcher class is responsible for catching the keywords that users want to search for. Then call DBManager to find the data in the database based on the keywords combined into a query statement.

Above all, we have designed the context of the diagram display task.

2) Structural Context :

In this section, we discuss the data flow processing by Figure 3. Meanwhile, we show the roles these objects play in the data flow processing, which describes the structural context of the Searching Data from Database task.

The user enters the keyword on the search page. Then the service combines the keyword into a query statement. DBManager searches the data from the database based on the query statement. Finally, the Searcher shows the data in the front end.

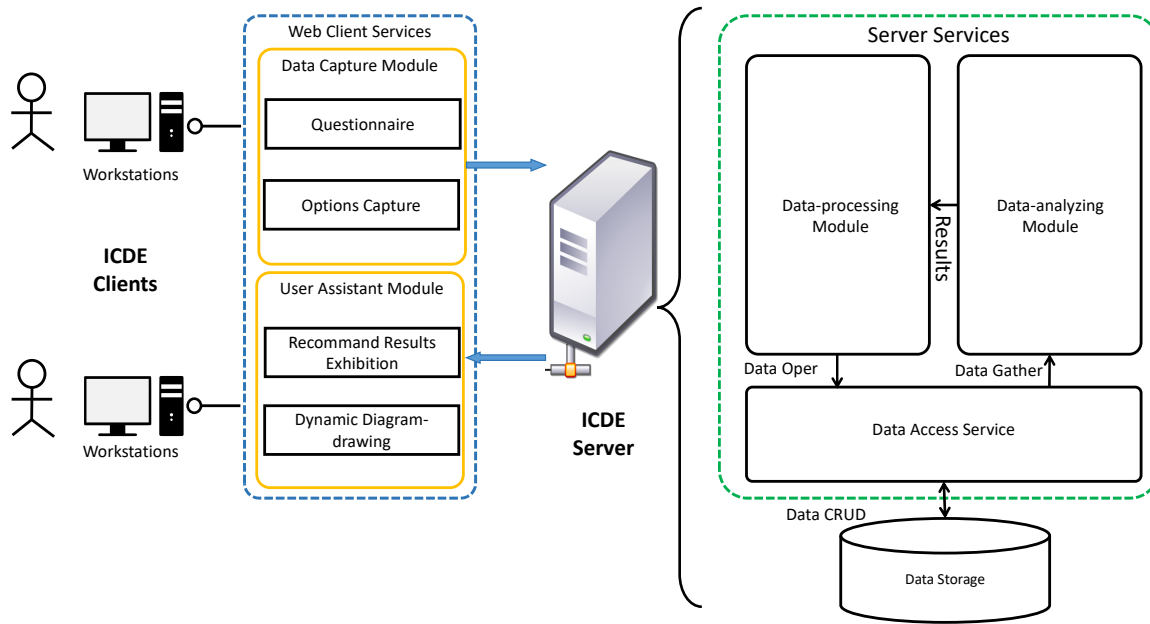


Fig. 1. The architecture graph of Tripedia.

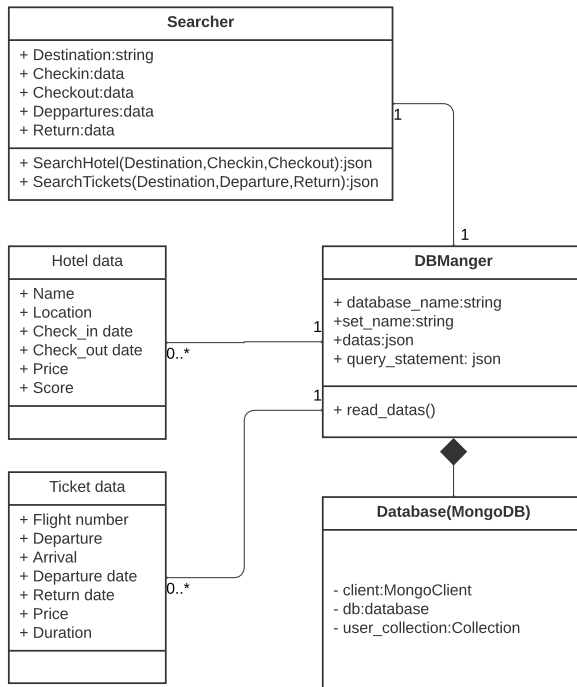


Fig. 2. Class diagram of Searching data from database

3) Interaction Conext :

The sequence diagram in Figure 4 shows the interaction context of this task.

The user inputs the keyword and calls the SearchHotel function or Search Ticket function of the Searcher object. The Searcher object calls the read_data function provided by the DBManager object to get the related data from the database.

In the end, the Searcher Object displays the data to the user.

4) Behavior Conext :

The state diagram in figure 5 shows the behavior context of this task.

The user enters the search web page. After inputting the keywords, the system searches data from the database. Subsequently, the website displays the data as a list.

B. Task: Crawler searching

1) Task Conext :

The class diagram in Figure 6 shows the relationships among the searcher, crawler and database manager.

The DBManager is a base class in the database interface, which encapsulates the Mongo Database interface. The DB-Manager can save hotel data and ticket data into the database.

The Searcher class is responsible for catching the keywords that users want to search for. When the searcher receives the

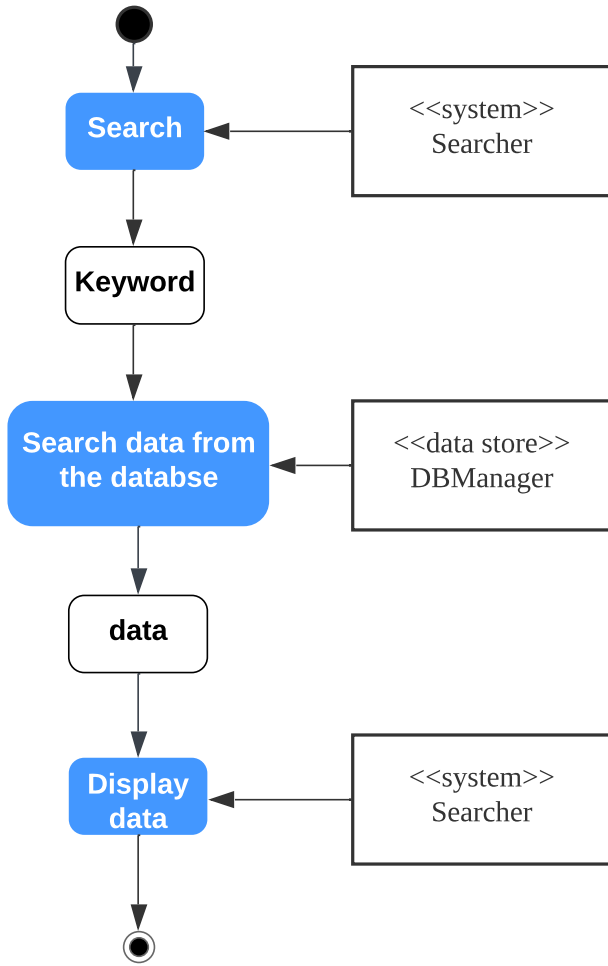


Fig. 3. Activity diagram of Searching data from database

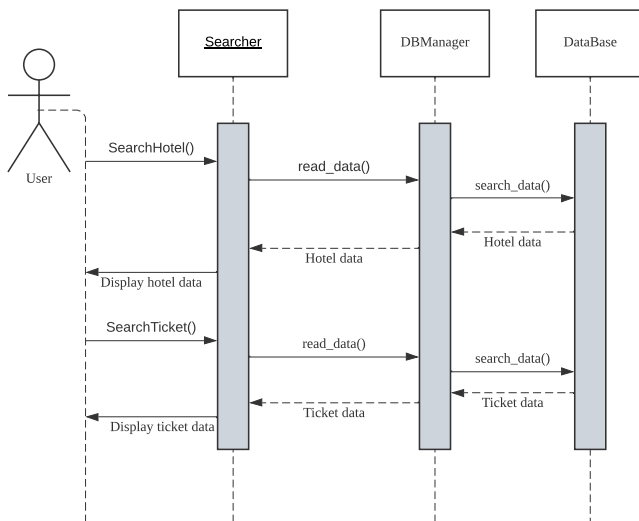


Fig. 4. Sequence diagram of Searching data from database

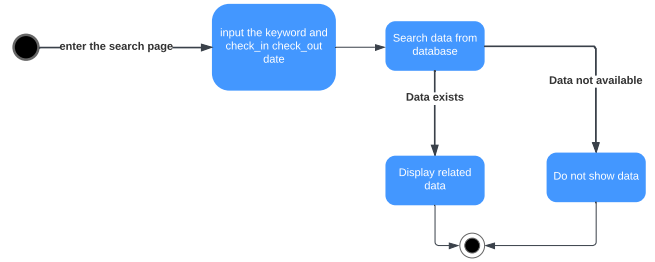


Fig. 5. Statement diagram of Searching data from database

keywords, it will call the Crawler to search information from the internet.

The Crawler class is responsible for catching data from the internet. The Crawler can be generalized as the HotelCollector and TicketsCollector. They have extended with some class functions related to searching hotel or ticket data from the internet. When they get hotel and ticket data, they will call the DBManager to save the data into the database.

2) *Structural Conext* :

In this section, we discuss the data flow processing by Figure 7. Meanwhile, we show the roles these objects play in the data flow processing, which describes the structural context of the Crawler searching task.

The user enters the keyword on the search page. Then the service calls the Crawler to search data from the internet. Finally, the Searcher shows the data in the front end, meanwhile, the DBManager saves the data into the database.

3) *Interaction Conext* :

The sequence diagram in Figure 8 shows the interaction context of this task.

The user inputs the keyword and calls the SearchHotel function or Search Ticket function of the Searcher object. The Searcher object calls the get_hotels pr get_tickets function provided by the Crawler object to get the related data from the Internet.

The crawler first gets the searchid from the website, and then uses the searchid to get the required data. When the Crawler receives the data, it will call the DBManager to insert it into the database and return the data to the Searcher Object.

In the end, the Searcher Object displays the data to the user.

4) *Behavior Conext* :

The state diagram in figure 9 shows the behavior context of this task.

The user enters the search web page. After inputting the keywords, the system will call the crawler to search data from the internet. Subsequently, the website displays the data as a list and saves the data into the database.

C. Task: The Analysis Diagram

1) *Task Conext* :

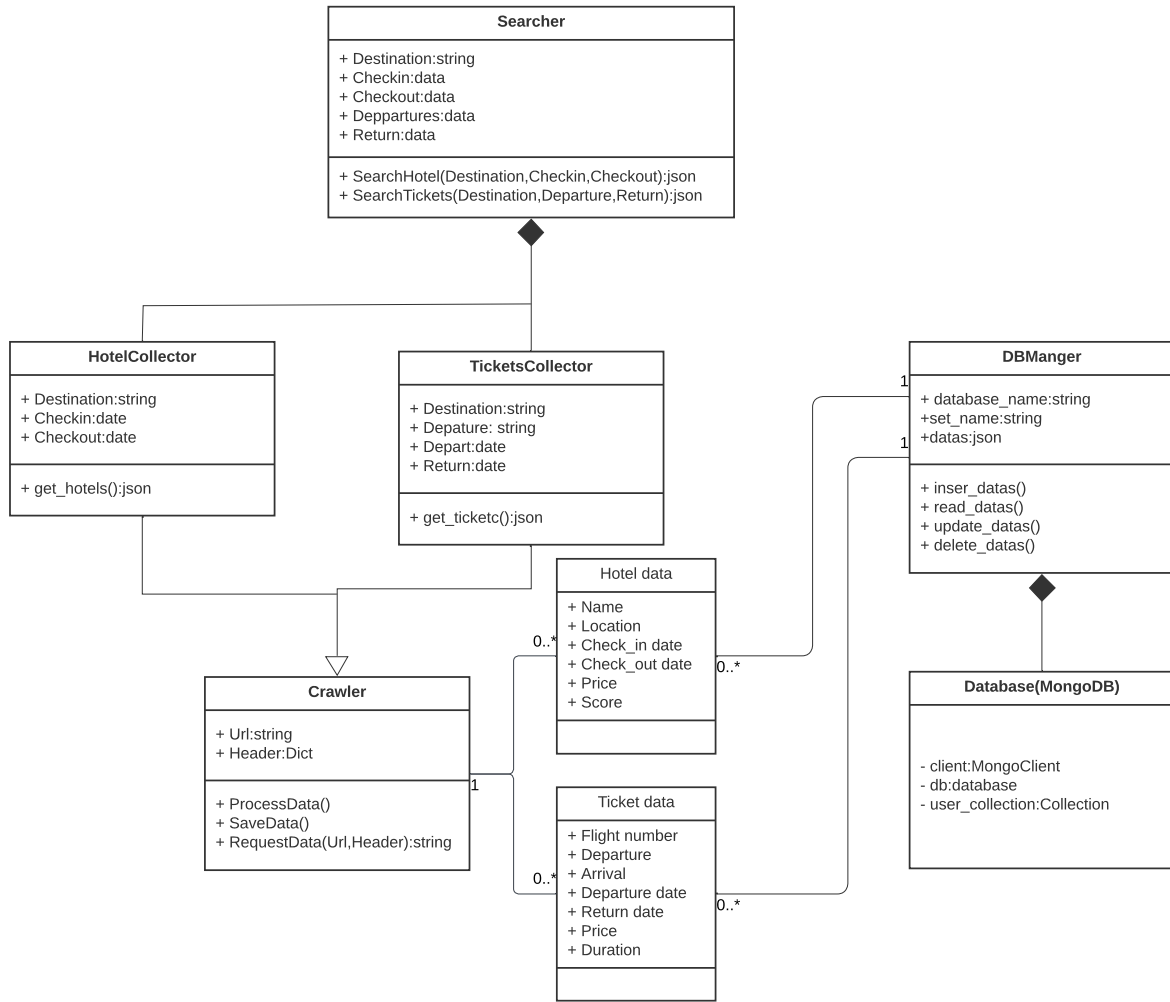


Fig. 6. Class diagram of Crawler searching

The class diagram in Figure 10 shows the relationships among the analyzer, questionnaire, displayer, and database manager.

For high efficiency and reusability, we designed a mediate layer for accessing the database. The MongoDBManager is a base class in the database interface, which encapsulates the Mongo Database interface. The Mongo DB Manager can be generalized as the Cities Information Collection and Preference Information Collection. The Cities Information Collection has extended with some class functions related to the city information. Meanwhile, the Preference Information Collection has extended with some class functions related to user preference information.

The Questionnaire class is responsible for catching user's preference information. A global Preference Infor Collection class instance provides the database interface for the data processing function in the Questionnaire class. Hence, the Preference Information Collection is associated with the Questionnaire.

On the other side of the class diagram, the diagram displayer performs the issues about analysis data displaying, which is based on the data analyzed by the KNNDataAnalyzer. The KNN Data Analyzer compares each city data entry with the user's preference and decides which city is the most appropriate recommendation. In this process, the KNN Data Analyzer calls the database interfaces provided by the Cities Information Collection and Preference Information Collection.

Above all, we have designed the context of the diagram display task.

2) Structural Conext and Task Data Flow:

In this section, we discuss the data flow processing by Figure 11. Meanwhile, we show the roles these objects play in the data flow processing, which describes the structural context of the Analysis Diagram task.

The user fills out the questionnaire on the questionnaire web page with the help of the questionnaire system. The Preference Information Collection saves the user's preference information for further analysis. Subsequently, the user en-

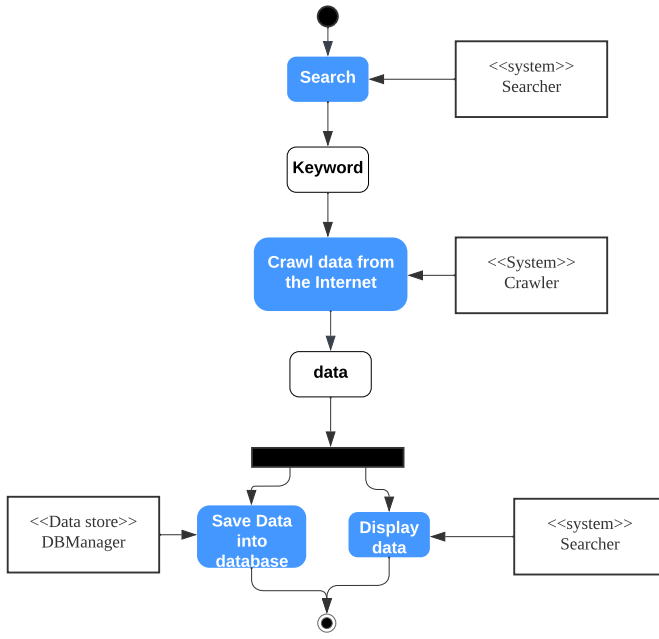


Fig. 7. Activity diagram of Crawler searching

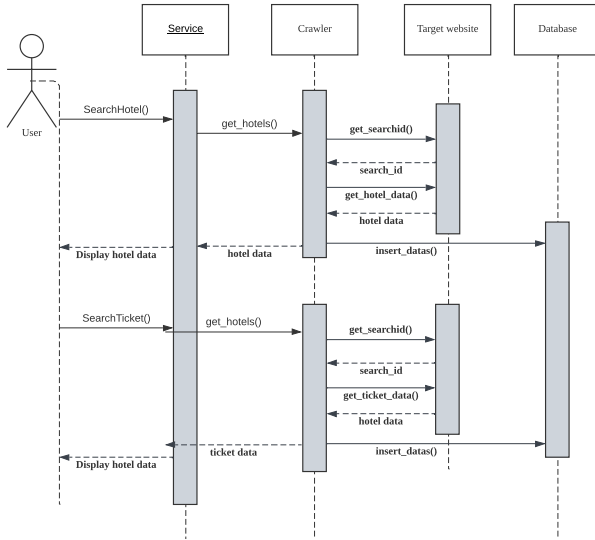


Fig. 8. Sequence diagram of Crawler searching

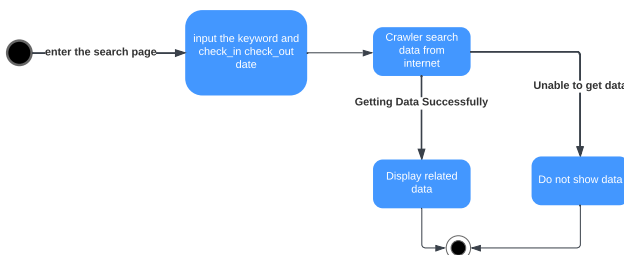


Fig. 9. Statement diagram of Crawler searching

ters the analysis page, which calls the Get Data Functions of Preference Information Collection and Cities Information Collection simultaneously. The preference data and city data are sent to the Data Analyzer for data analysis. Finally, the Diagram Displayer shows the data diagram based on analyzed data.

3) Interaction Context :

The sequence diagram in Figure 12 shows the interaction context of this task.

The user fills out the questionnaire and calls the PostQuestionnaire function of the Questionnaire object. The Questionnaire object calls the StorePreferenceData function provided by the Preference Infor Collection object to save the preference data into the database.

Then the user clicks the 'recommend' button on the home page and calls the EnterAnalysisPage function, which is provided by the Diagram Displayer object. The Diagram Displayer object makes a request to the Data Analyzer for the recommendation information. The Data Analyzer gets the user preference data from the Preference Infor Collection object according to the UID. Meanwhile, all city data is gained from the Cities Infor Collection object. The Data Analyzer processes a recommendation algorithm based on these data and feeds back the recommended city information and popular city information. The Diagram Displayer shows the data diagram and returns an OK message to the user.

4) Behavior Context :

The state diagram in figure 13 shows the behavior context of this task. The state transition is simple and most functions in this task process data invisible to users. We divide the state according to the function of the web page. In this section, we only care about the states related to the analysis diagram task.

The user enters the questionnaire web page, the system enters into the Questionnaire State. The Questionnaire State displays the multiple psychological questions for users. After finishing the questionnaire, the system saves the preference data into the database. Subsequently, the website comes back to the home page and the system enters the home state. When the user clicks the 'recommend' button, the website jumps to the Analysis Diagram Page. After the system completes the recommendation, the Analysis Diagram Page displays the data diagram.

IV. UNCOMPLETED TASK DESIGN

1) Task Context :

2) Structural Context :

3) Interaction Context :

4) Behavior Context :

V. PROGRESS EVALUATION & ANALYSIS

Class Diagram

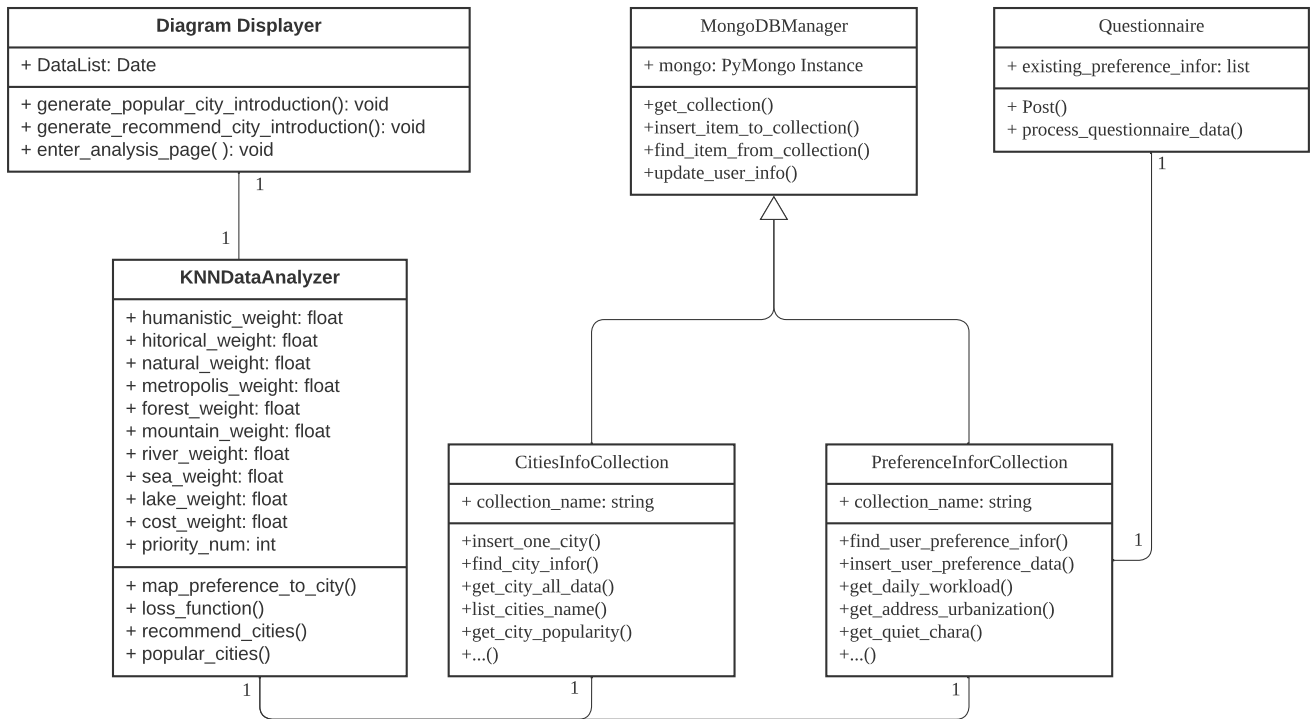


Fig. 10. The class diagram of analysis diagram task.

Structural Conext & Data Flow Activity Diagram

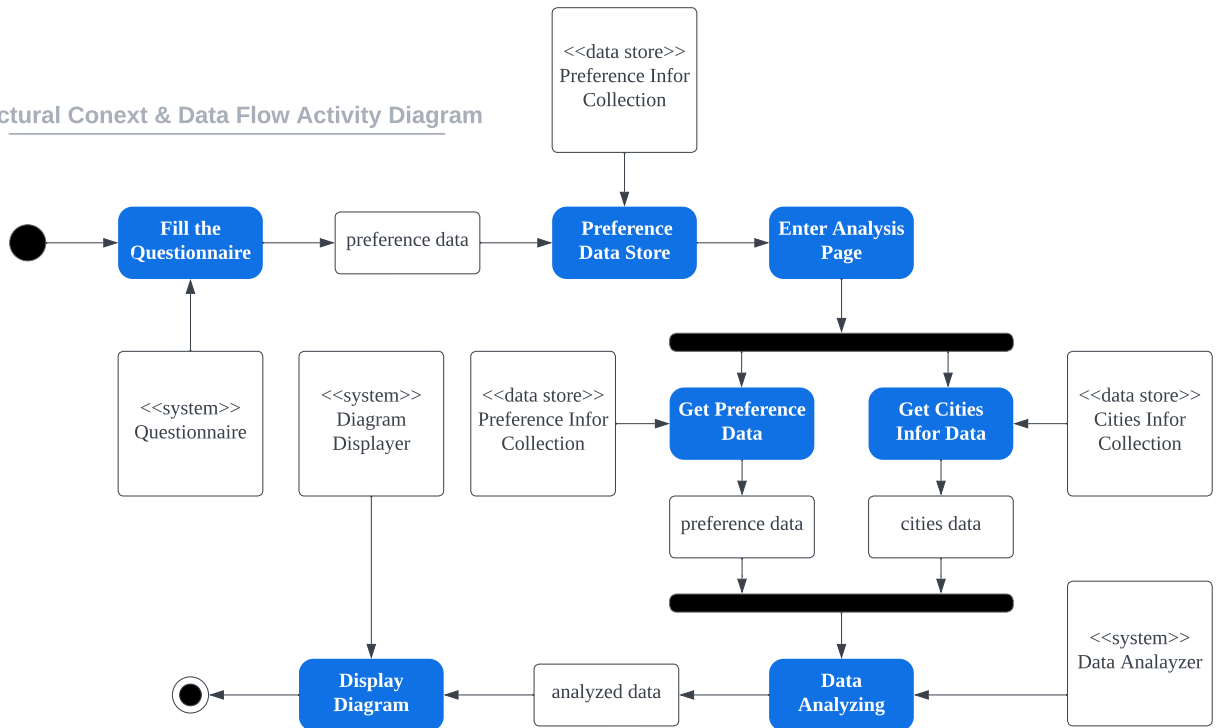


Fig. 11. The structural context and data flow of analysis diagram task.

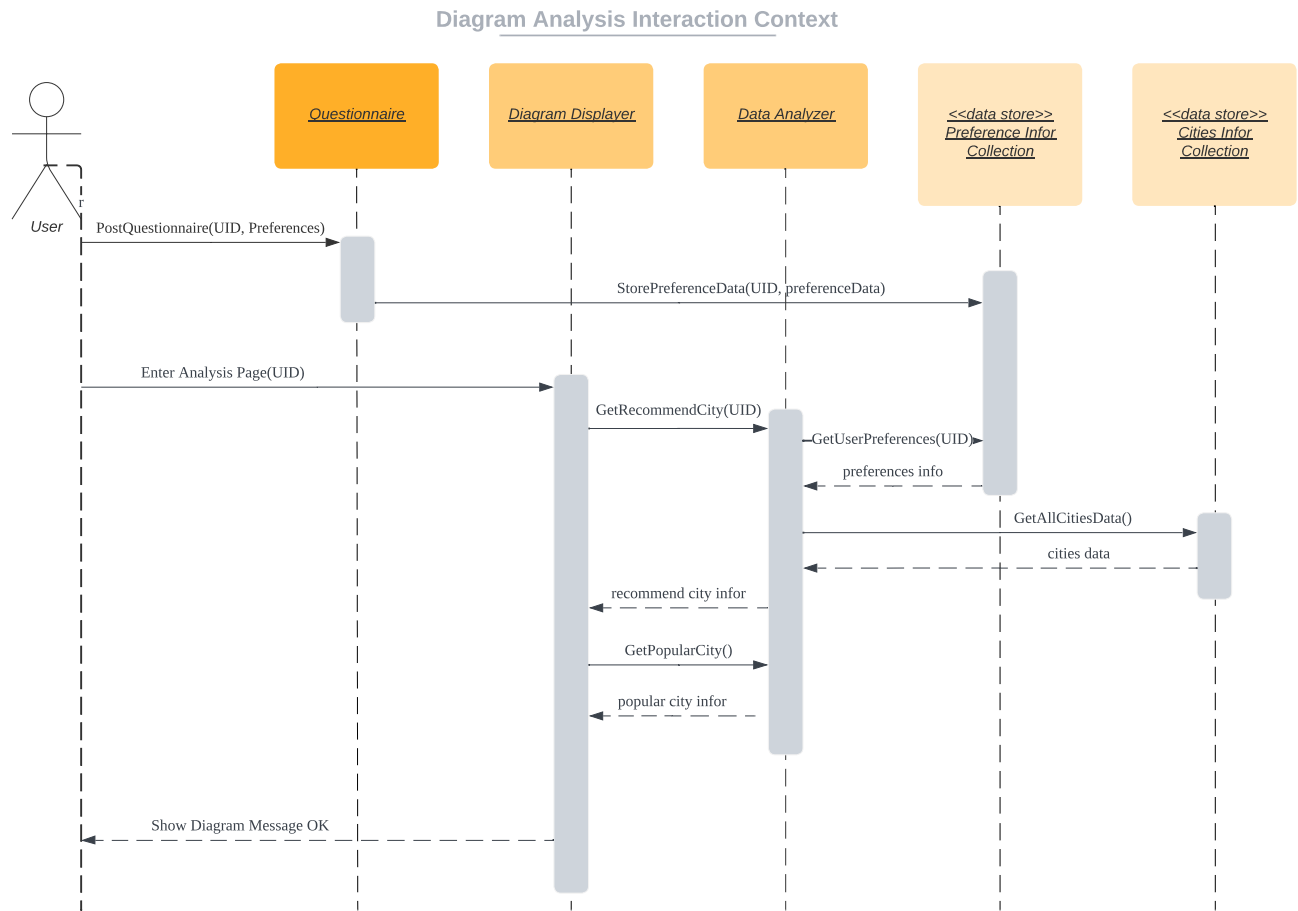


Fig. 12. The interaction context of analysis diagram task.

Analyze Diagram State Diagram

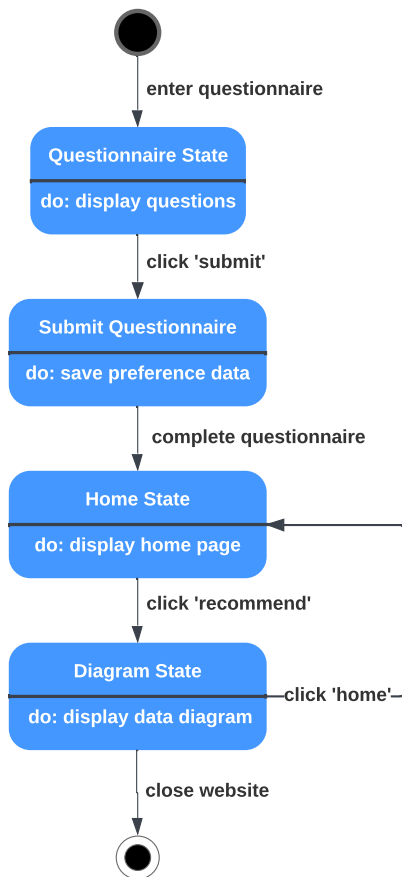


Fig. 13. The state diagram of analysis diagram task.