**Overview**

**a. Describe your understanding on problem statement**

1. With the increasement of business teams, the business data structure becomes complex and diversified. At present, each business team will maintain its own query system, and the cost is very high. Also, various teams have done a lot of repetitive work.
2. When users want to execute the same or similar queries, they need to manually execute the same process every time, which cannot be processed concisely and intelligently.
3. If one-time data needs to be obtained in a specific scenario, the user may be directly given access to the database, and there is no authorized audit process, so the data security cannot be guaranteed.

**b. Team members**

Zoey：Captain of the team. A female front-end developer integrating appearance and ability. Mainly responsible for user function interaction and page module planning and development.

Clay: The boy who is very cheerful, sunny and handsome. The system analyst and back-end developer of our team. Be responsible for sorting out the main functions of the system and the development and deployment of core functions.

Yunpeng：A lovely boy who graduated two years ago. He is serious, responsible and thoughtful. The system designer and back-end developer of the team are responsible for the design of core business processes and the development and testing of core functions.

Wuyi: The captain's cat is the mascot of our system. Mainly responsible for providing spiritual comfort.



**High Level Solution Approach**

1. **Describe Solution**

In order to solve the high cost of data query and export due to multiple business teams and multiple data sources, and the problem that the management is not simple and convenient. So we plan to develop a general data query system.

The system defines a complex query process that can be reused, dynamically changed and configured with different data sources. The key of the system is how to define a concise, customizable dynamic condition and reusable query process, and make this process adapt to the specified user-defined data source.

The highlights of this system are as follows,

1. Elaborate definition of query process

In order to make users focus on maintaining the query process instead of repeatedly creating and configuring each time, we will make each query process of users reusable and set dynamic conditions at the same time.

1. Conduct safety audit on query process and operation management

* The query process definition and modification of the system are protected.
* The modification of each query will archive the operator and operation time for subsequent audit.
* Each query execution will go through permission and security verification. Query results can also blur sensitive data.

1. Customer defined data source information

This project can adapt to a variety of data sources and support users of different business teams to configure different data formats on the interface.

1. Brief but effective UI design

Our system is very convenient to use. It has data source configuration, query display, query configuration, query execution records, query report and download page. The page jump logic is simple and clear , which ensures that all functions are easy to use.

1. Query can be executed repeatedly

Each query can be configured with periodic execution, which can be executed in the background according to the cycle and save the query results. The query criteria executed in the background can also be dynamically configured. For example, it supports exporting the product sales summary information of the previous day at 8 a.m. every day.

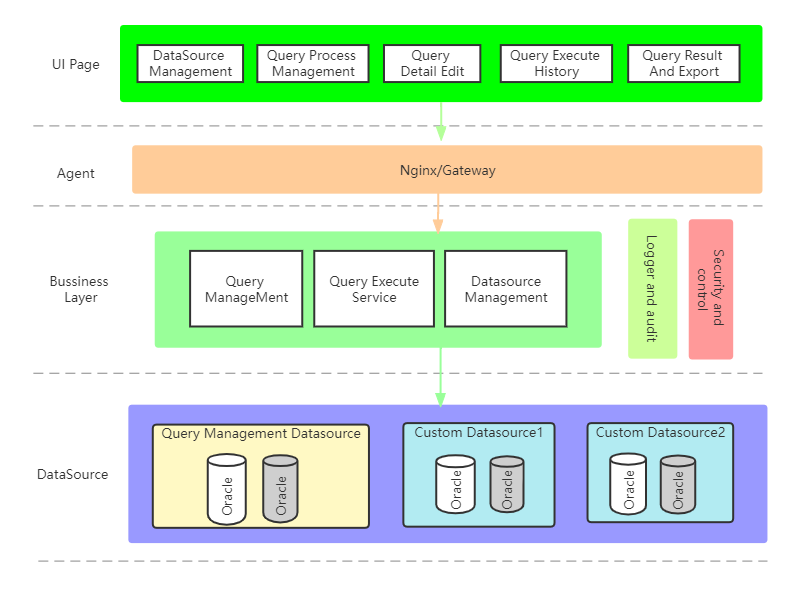
1. **High Level Architecture**

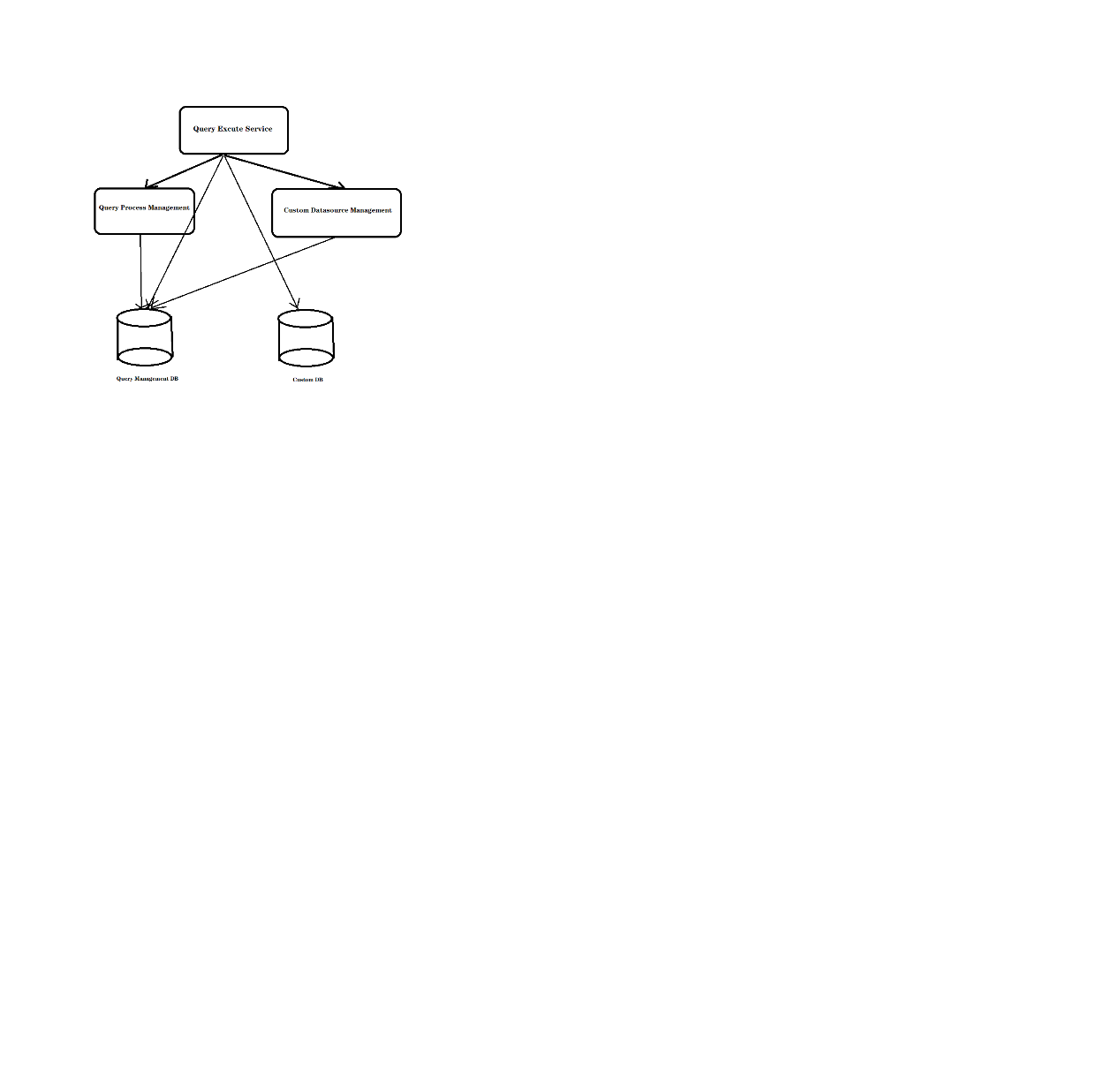
We have five functional pages, with three independent back-end modules supporting the main functions.

UI functional pages

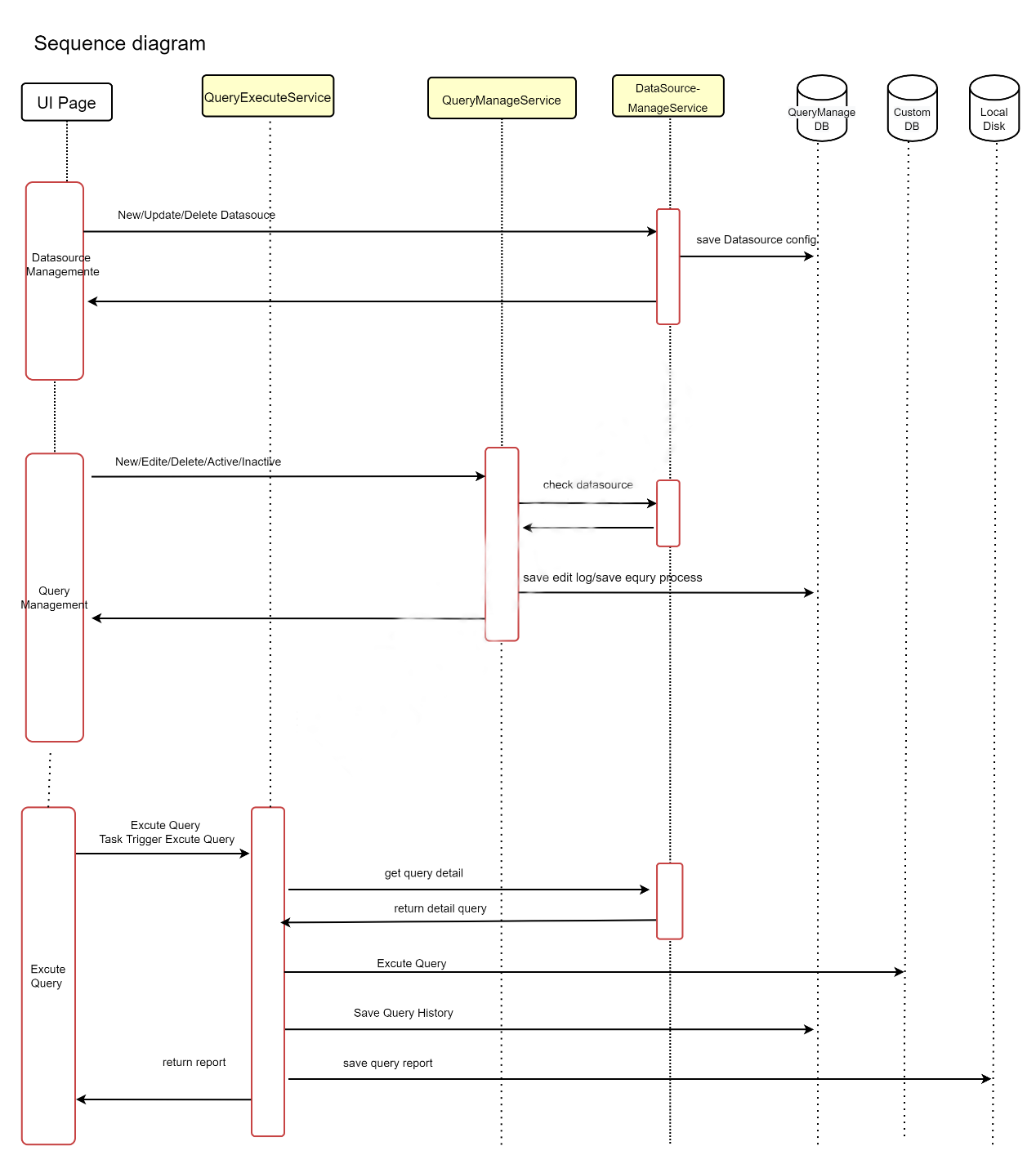
* 1. Data source configuration page
  2. Query process list display and control page
  3. Detailed configuration page of the query process
  4. Query execution record page
  5. Query report review and download page

1. System structure diagram and functional sequence diagram

2.1 System structure diagram： 

2.2 The dependencies between the backend services are as follows： 

2.3 Functional sequence diagram



**Impact**

1. **Product Approach**

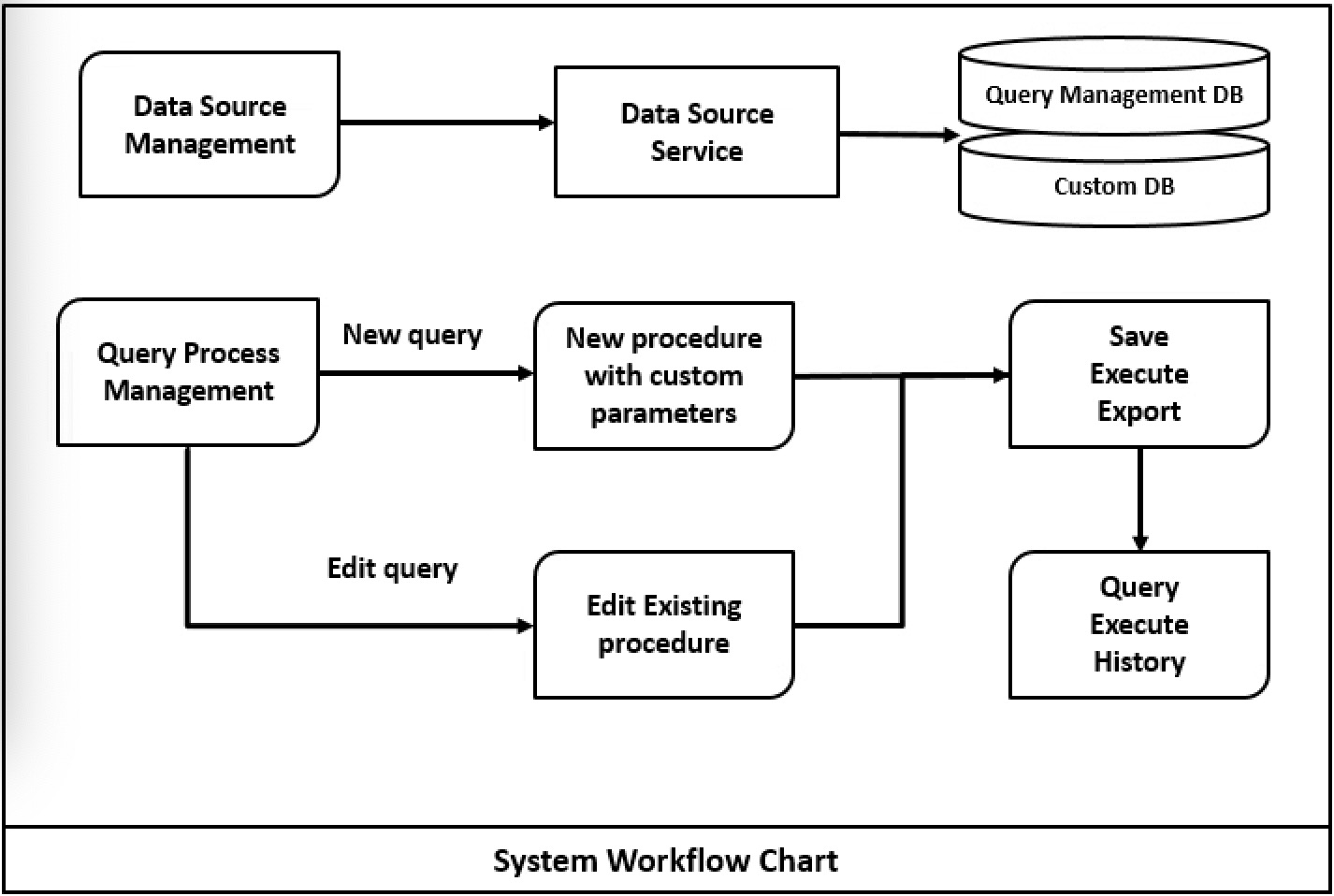
1. Firstly, user should configure the data source information on the [Data Source Configuration] page, and need to enter key information like database connection information, table name, field information, belonging project, etc.

2. Then user can browse the query information on the [Query Management] Page, including the previously created query process, and performs operations such as activation, shutdown, deletion, etc.

3. When user on page [Add or Modify Query Details], they need to specify the query target table, query fields, input static conditional statements or dynamic conditional statements, and specify the execution cycle and other information.

4. User can also choose to copy their own query based on the existing query, and set the dynamic conditions if they want. For example, there is already a summary of the weekly sales report of product A that is executed by query one every Monday. At this time, if you need to obtain the daily sales report of product B on the previous day, you can copy query one and modify the query process slightly.

5. User will be allowed to query the actual parameters used in the historical execution of the query and the execution progress on the [Execution Records] Page.

6. User can click execution history to enter the current [Execution Result] Page, to view the current execution result, and export the query result data. 

1. **Business Impact**

It is recommended that the business services which will use our system need to set up a separate database read-only account.

The use of this system will have no impact on each business services, and will not modify any of the existing business data.

**Non-function Requirements**

1. **Scalability**

Our system support different data formats, therefore different team with various data sources can use our service directly.

we have planned access the interface of machine learning and artificial intelligence in the identification of PII data, so as to make the system more Scalable.

**b. Throughput**

The business functions have been vertical split into three services to improve throughput. First of all, the management of customer data source is a separate module. In addition, query process management and query execution are divided into two modules. Query management itself will not impact with query execution.On the other hand,the thread execution pool will be used to maintain the performance of the query process.

1. **Security**

After the user logs in, the token is used throughout the process. The management operation of query will verify the user identity.

The key operations of the query will keep the user's operation log.

And we will not display sensitive data in all businesses .

**d. Entitlement**

The authorization part is supported by third-party services, and in the future, we plan to add different roles and permission control to make the system permission management more clear.

**e. Cloud Deployment**

The deployment of this system is simple. The front-end Angular project will compiled into a static HTML project, which can be directly placed in a static resource server like nginx. The back-end will be packaged into three jar packages, which can be deployed to the cloud server with a set of database environment.

**f. Test Automation**

The system will provide comprehensive unit test cases to ensure the functional reliability of the system, but the automatic test module will not be introduced into the system at this stage.