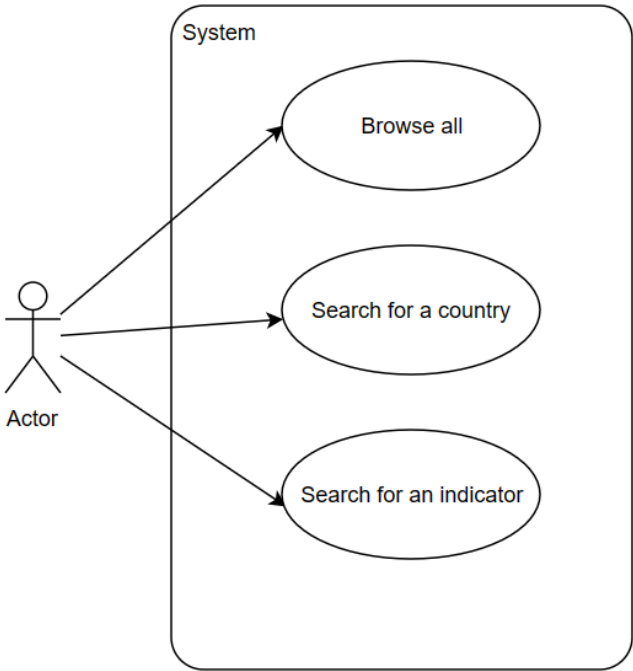


Requirements

Elicit the requirements

People's quality of life and living standards have been improved today, and today, with data, people have begun to pay attention to numbers and explore the truth behind the numbers; At the same time, it is more acceptable to plot the data into a variety of charts. Finally, in the context of big data, data screening is also very important. The above ideas form the following use case diagram:



Document the requirements

Based on the above considerations, our web app has the following requirements:

Name	Audience	description
See all data	Visitors, data analysts	View all the data for a dataset in tabular form.
Search for a country	Visitors, data analysts	Search for all information about a country, presented in a table.
Search for an indicator	Visitors, data analysts	Query the changes of a metric and display it in the form of a line chart.

Name	Audience	description
Query data for a country for a specific year	Visitors, data analysts	Query the data of a certain country for a certain year, and display the recorded data in the form of a bar chart.

Prioritise the requirements

After the four requirements were determined, since there were not many requirements, no specialized technology was used in the confirmation of priorities, but a simple analysis. First of all, browsing all the data should be the most important requirement, and it is also the easiest to achieve the need, from the target, get all the data can be analyzed by themselves, for developers, full-text search may be simpler. The second is the drawing of various tables, and people prefer to look at pictures rather than numbers. Finally, it is necessary to look at all the information of a country, so the final priority order is as follows: (the lower the number, the higher the priority)

Priority	Name
1	See all data
2	Search for an indicator
3	Query data for a country for a specific year
4	Search for a country

For personal reasons, I plan to build a simple web app with a separate front and back end, while using a database to store my data sets. In terms of interface, it adopts the required RESTful style API, and adopts MVC architecture in the back-end design. Below I will go through the techniques I use one by one.

1. Streamlit

Streamlit is a Python framework for machine learning, data visualization, and it can build a beautiful online app in a few lines of code. Streamlit supports the display of many controls, and for the current job, its easy to learn and use features made me decide to use it to complete the front-end code.

2. Flask

Flask is a lightweight, customizable framework written in Python that is more flexible, lightweight, secure, and easy to use than other similar frameworks. It can be well combined with the MVC pattern for development, and the implementation of feature-rich small and medium-sized websites or Web services can be completed in a short time. Most importantly, Flask supports the design of RESTful APIs very well, making interface design simple.

3. SQLite

When it comes to database selection, SQLite is definitely the best solution. SQLite, a lightweight database, is an ACID-compliant relational database management system, it is contained in a relatively small C library, he is very good at learning database programming, improving programming ability, etc.