**4-4 Journal Software Application Requirements**

The client-server model uses the advantages of hardware to reasonably distribute tasks to the client and server. The client submits a request to the server, the server receives the request, processes the request, and returns the result to the client.

Server side: The client sends a request with data to the server through the endpoint, and the server returns an appropriate response. Usually, the server receives the request through these instructions. GET: Get resources from the server, POST: Save resources on the server, PATCH or PUT: Update existing resources on the server, DELETE: Delete existing resources on the server.

Client side: In order to allow games on multiple platforms to be used on the server, the first thing developers need to do is compatibility testing. Secondly, developers need to ensure data synchronization across multiple platforms, which can be achieved through API. Collect data on the web and app, transmit the collected data back to the server through the API, and then transmit it back to the client through the HTTP protocol after processing. To add new users, first make sure that the account and password are unique login information, then send a new request to the server, and then add new users after server verification. Therefore, in subsequent game applications, identity authentication functions should be added to ensure that user information will not be stolen and that there will be no duplicate logins. If we need to add other platforms such as xbox, developers need to understand the development tools of xbox, and then develop new apps through the development software of that platform.