**Performance:**

* RESOURCE DEMAND:
  + Manage event rate: Reduce sampling frequency to the system, to reduce the event rate to the system
  + Bound execution times. Place a limit on how much execution time is used to respond to an event
* RESOURCE MANAGEMENT
  + Introduce concurrency. If requests can be processed in parallel, the blocked time can be reduced: Use two WEB server to balance request
  + Increase available resources. Faster processors, additional processors, additional memory, and faster networks all have the potential for reducing latency.
* RESOURCE ARBITRATION
  + First-in/First-out. FIFO queues treat all requests for resources as equals and satisfy them in turn

**Security:**

RESISTING ATTACKS

* Users want to log into the system will need to use one account and a password. Only accounts with the correct password can access the system.
* Authenticate users: Verify the system's users. Administrator will be created for an account and password for each user to be granted access to the system. Each different users will have different rights in the system, working with different data, this right will be confirmed when a new account
* Authorize users: Each account will have a different right in the system
* When logging into the system, account information and passwords will be secured
* Maintain data confidentiality: using transmission Secure Sockets Layer (SSL) for a Web-based link to security information and account passwords
* When user change information in the system, system will record it in to log

**Availability**

* **FAULT RECOVERY**
  + Passive redundancy: Use backup database and primary database, when primary database crash, system can operation normally with backup database