# Load Balancing

A load balancer sits in front of your servers, acting as a "traffic cop," directing client requests across all servers capable of satisfying those requests in a way that maximizes performance and reformulation testing while ensuring that no single server is overworked, potentially degrading performance. The load balancer transfers traffic to the remaining online servers if a single server goes down. When a new server is added to a server group, the load balancer begins sending requests to it automatically.

# API / Middleware Justification

# Software Components used

# Risk Assessment

|  |  |
| --- | --- |
| Risk | Management Strategy |
| Data loss/technical failure | Backup and copy on a storage device. |
| Overrun on the schedule | The project plan now includes a contingency plan. Highlights and supervisor meetings will be used to track and monitor progress. More days have been added to the estimated completion dates for each step. |
| Complexity of the project | To adhere to a set of rules that would allow the basic functionality to be implemented and a working product to be produced. |
| Problems with the technology needed to complete the project | A basic prototype will be constructed for each phase to examine and validate validity. Maintaining a record of each stage. |
| Other requirements | There will be other projects operating concurrently, and this will receive full attention. A time management strategy would assist me in staying on track. |
| Lack of information on this topic | * Before starting, do a feasibility study on the topic. * Read books, essays, etc. on the topic. * Research on the internet. |
| Bad Timing | * Plan, then stick to it. * You must determine which project stages or components are most important to you, as well as the short, medium, and long-term impact of each stage/component. |
| Poor Code Quality | * Review of the code * Coding rules and guidelines that are easy to understand * All code is tested. * The Method of Work. |