Contents

1 Non Functional Requirements

1.1 Performance requirements

1.1.1 Reasons for performance as a non-functional requirement

- Performance is one of the core components of what is required from the project management system, it needs to be effective and efficient.
- Performance can play a large role in usability, usability describes the users experience well using the product. The project management system must deliver a positive user experience.

1.1.2 Strategies to achieve this non-functional requirement

- The system needs to be able encompass over 200 concurrent connections.
- The DB needs to handle over 1000 requests and responses each second.
- The latency associated with verifying a users log in credentials must be fall under 200ms.
- The latency associated with requesting and receiving a response for an accessible web page must fall under 50ms.

1.1.3 Patterns to achieve these strategies

- Expert Pattern
- Layer pattern

The Expert pattern can be used to direct requests to different locations. The pattern ensures your program utilizes efficient resource management by directing requests and responses to the respectful parts of the system that can Handel them correctly. This relieves bottle necking a single part of the system with all responsibilities. Utilizing the layer pattern allows us to divide our system into subsystems which can be run in separate locations.We can divide our initial system into A front end, a back end and a database manager. By splitting our system up, we can alter how many resources we allocate to each system, thus we utilize our recourse's efficiently and provide loose coupling between the different parts of the system.
By combining the expert and layer pattern, we can handle different requests efficiently and correctly with their associated subsystem.

1.2 Quality requirements

1.2.1 Reasons for quality as a non-functional requirement

- By stating Quality requirements, you define the desired attributes which your program must have. Defining your programs attributes moulds key factors of its development and sets quality milestones and goals.
- When the project is ready for deployment, the quality requirements which were set at the begging of development define how reliable the end product is or should be.

1.2.2 Strategies to achieve this non-functional requirement

- When Queried to allocate a team, the resource allocation algorithm must produce one within a maximum of 10 seconds.
- The system shall be available 99% of the time, only going down in the duration of maintenance or while implementing updates.
- When queried for a project team and members are available, the resource allocation algorithm should never respond with no allocated team members.

- When queried for a project team by a project manager, the resource allocation algorithm should provide a compatible and satisfactory project team to the project manager 85% of queries made.

1.2.3 Patterns to achieve these strategies

- This will depend on the pattern used by the artificial intelligence resource allocation algorithm.
- layer

Using the layer pattern we can divide the responsibilities of quality to the respective three layers. Each layer handles its own quality management, this promotes loose coupling regarding quality dependencies between subsystems. Improvements can be made to separate layers independently to satisfy the systems quality requirements.

1.3 Security requirements

1.3.1 Reasons for Security as a non-functional requirement

- Kpmg offers services in both data analysis and auditing. Both areas deal with highly confidential client information and regard security as top priority. The project management system may not have anything to do with these services, however it is a representation of kpmg's ethics and it must value security regarding client information with the same importance.
- If the project management system is to be hosted from the same physical servers as other company systems, it must have no area for exploitation. Security requirements ensures this.
- confidential information regarding projects should have no opportunities to be exploited to the public via malicious attacks on the system, security requirements protect the system from letting this happen.

1.3.2 Strategies to achieve this non-functional requirement

- There is a hierarchy of users for the application. The range includes administrators, directors, project managers and further employees below these roles. Each user has different responsibilities and authority over the project which gives them project management options to govern the project that are not available to other users. Access to options available to users with a certain role should only be available to these users, session and verification management of the system will ensure this.
-

1.3.3 Patterns to achieve these strategies

- Chain of responsibility

Using the chain of responsibility pattern, we can create a chain of handlers for our requests and responses. The client can send a request to the server which either handles the request or sends it to the database manager whereby the response is then sent back up the chain. This allows us to have greater control over the security of the system as a whole by carefully checking and handling each request at different stages before it can access any information from the database.

1.4 Interface requirements

1.4.1 Reasons for Interface as a non-functional requirement

- Interface requirements cover a broad spectrum including the usability of the system as well as its hardware and software interfaces and their respective capabilities.
- The usability of the system defines the user experience well using the system. To elicit a positive experience, the system must take into account and deploy many usability guidelines. KPMG wish to update their current project management system with a new design due to the lack of usability effectiveness of the current system.

1.4.2 Strategies to achieve this non-functional requirement

- The new system's front should be developed with usability as its key target. its interface should have a modern and intuitive design which brings about a positive experience to the user using it.
- All subsystems should communicate with one another through http Get and Post requests.

1.4.3 Patterns to achieve these strategies

- layered pattern