**Non-Functional Requirements**

* **Usability**
  + It considers the ease of use and learnability of this game-based educational platform. As a software engineering point of view, usability is the degree to which the software of the platform can be used by students and teachers to achieve the objectives of the project helping the students to understand basic educational concepts while they will be playing simple games in an effective, efficient and satisfied context of use.
* **Reliability**
  + It emphasizes **dependability** in the lifecycle management of the platform where dependability, or reliability, describes the ability of the system to function under the stated conditions for a twenty-four hours per day. Reliability may also describe the **availability** where the system has the ability to function at any time. Reliability is theoretically defined as the probability of success (Reliability = 1 – Probability of Failure), as the frequency of failures; or in terms of availability, as a probability derived from reliability, **testability** and **maintainability**. Testability, maintainability and **maintenance** are often defined as a part of "reliability engineering" in Reliability Programs. Reliability plays a key role in the cost-effectiveness of the systems.
* **Performance**
  + It considers the amount of work accomplished by the system. And since the system has high performance, it involve the following:
    - Short response time for a given piece of work: 0.1 to 1 second
    - High throughput (rate of processing work)
    - High availability of the computing system or application
    - Fast (or highly compact) data compression and decompression
    - Short data transmission time
* **Supportability**
  + It refers to the ability of [technical support](https://en.wikipedia.org/wiki/Technical_support) personnel to install, configure, and monitor the product, identify exceptions or faults, [debug](https://en.wikipedia.org/wiki/Debugging) or isolate faults to [root cause analysis](https://en.wikipedia.org/wiki/Root_cause_analysis), and provide [software maintenance](https://en.wikipedia.org/wiki/Software_maintenance) in pursuit of solving a problem and restoring the product into service.
  + Examples of features that facilitate serviceability include:
    - Network monitoring
    - **Documentation**
    - Software upgrade
* **Reusability**
  + It considers the use of this platform assets which include code, software components, test suites, designs and documentation where they are in some form within the software product development process.