**Experiment 03- Mapping of Non-functional Components with System Requirements**

**Learning Objective:** To state the non-functional requirements of the hangman game.

**Tools:**  MS Word

**Theory:** Non-functional requirements are specifications that define how well a software should perform. They cover areas such as speed, security, reliability, and usability. Unlike functional requirements that focus on what software must do, non-functional requirements determine how effectively it should do it. These requirements ensure that the software meets quality standards and user expectations. Factors like measurement scenarios, workload, architectural constraints, and scalability expectations must be considered when documenting non-functional requirements. They are crucial for creating software that performs well and provides a positive user experience.

**1. Security –**

**Authentication**: We implement secure user authentication methods for both creating and joining lobbies by using passwords which are randomly generated.

**Authorization**: Only authorized users (those who have entered the correct password) would be allowed to join the game lobby.

**2. Performance –**

**Low Latency:** The application should have minimal delay in processing user inputs and displaying game updates. To achieve this we will use UDP connection.

**Fast Loading Times**: The game should load quickly, including the initialization of the application and the loading of game sessions.

**3. Scalability –**

**Horizontal Scaling**: The game server architecture supports horizontal scaling to accommodate an increasing number of players. Thus a lobby with a large number of players can be created which would enhance the happiness index and competitiveness.

**Load Balancing**: The connections are distributed evenly among available servers. Thus there can be more than one lobby created.

**4. Maintainability –**

**Modular Codebase**: The code structure is divided into different classes and functionalities

**Version Control**: The changes made to the code is frequently updated on GitHub which allows to get hands on experience on version control

**5. Portability –**

**Cross-Platform Compatibility**: The game runs on multiple operating systems without significant changes. Thus we can have one user playing on Linux and other on Windows

**6. Usability –**

**Intuitive UI**: The UI of the game is simple and user friendly

**Consistent Experience**: Players will have a consistent user experience across different devices and screen sizes. Hence they will be able to enjoy the game anytime, anywhere.

**Learning Outcomes:** The student should have the ability to:

LO 1: Understand non functional requirements

LO 2: Identify & elaborate the various non functional requirements

**Course Outcomes:** Upon completion of the course students will be able to understand and explain the different non functional requirements needed for their project.

**Conclusion:**

From this experiment we were able to understand what non-functional requirements are, identify the different types and hence were able to elaborate on non-functional requirements used in our project.

**For Faculty Use**

| **Correction Parameters** | **Formative Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning Attitude [20%]** | **Total** |
| --- | --- | --- | --- | --- |
| **Marks Obtained** |  |  |  |  |