**Project Non-Functional Requirements:**

**GitHub Wiki Link:** [**https://github.com/djgamekid/GDP-Group-I-bearcatmanager/wiki/Non%E2%80%90Functional-Requirements-List-(Iteration-2)**](https://github.com/djgamekid/GDP-Group-I-bearcatmanager/wiki/Non%E2%80%90Functional-Requirements-List-(Iteration-2))

Darren Ross, Anthony Jack, Manikala Chevitipalli, Sai Achyuth Konda

**Project Brief Description**

***Bearcat Manager*** *– A Web application that hopes to create an application that can be used to create and manage events and allow users to get tickets to events*.

**Non-Functional Requirements List:**

**Security**

1. Encryption: All login information will be encrypted in the database, ensuring user credentials are secure and inaccessible to unauthorized personnel.
2. Two-Factor Authentication (2FA): Email-based 2FA will be implemented to provide an additional layer of security, verifying user identity through a secondary method.
3. Access Control: The database will have restricted permissions, ensuring only authorized users can access sensitive information.

**Scalability**

1. Event Archiving: The system will maintain an archive of past events, enabling easy reintroduction of recurring events.
2. User Notification: The system will be capable of notifying all users about new potential events, accommodating an increasing number of users without performance degradation.

**Usability**

1. Simplistic Structure: The website's design will be straightforward and user-friendly, allowing easy navigation across different pages.
2. Interactive Design: Users will experience fluid navigation through the site, aided by interactive buttons and small animations.

**Performance**

1. Real-Time Synchronization: The system will synchronize data in real-time, ensuring that any changes made by users or admins are immediately reflected.
2. Concurrent Handling: The site will be optimized to handle multiple user requests simultaneously, ensuring smooth performance under load.

**Interoperability**

1. Data Format Support: The system will support common data formats (e.g., JSON, XML, CSV), facilitating seamless data exchange with other systems.
2. RESTful APIs: APIs will adhere to RESTful standards, ensuring efficient interaction with external systems.
3. Secure Authentication: The system will support OAuth 2.0 or equivalent secure authentication methods for safe integration with third-party systems.

**Portability**

1. Platform Independence: The system will be designed to run on various operating systems, including Windows, Linux, and macOS.
2. Deployment Flexibility: The system will be deployable across different environments, such as on-premises servers, cloud platforms (e.g., AWS, Azure), and hybrid cloud environments.
3. Containerization: Deployment via containerized solutions (e.g., Docker) will be supported to ensure consistency across development, testing, and production environments.

**Stability**

1. Error Handling: The system will include robust error handling to maintain stability and prevent crashes during unexpected situations.

**Maintainability**

1. Code Documentation: Comprehensive documentation will be maintained for the codebase, facilitating easier updates and modifications.
2. Modular Design: The system will have a modular design, allowing individual components to be updated or replaced without affecting the entire system.