SYSTEM REQUIREMENTS

Our system will let the user experince wrestling management, and will form game flow, produce results according to user preferences and strategies. While these processes system will use instant user choices and preferences, datas from database.

User choices will include wrestler transfers, trainings, team strategies and preferences. Match results will be produced after processing all these choices. All these features of the game are the parts of our system which has several functional and non-functional requirements.

1. Functional Requirements

Data Inputs To System
Since our system will process preferences and strategies fetched from database or user, data inputs are the main requirements of our system. They will be used to produce results and to shape the flow of the game.

Process To Do At Each Screen
Work flow will be provided from user preferences at 12 different layouts which have already been designed. User's activity at the layout side will trigger the processes to be performed.

Workflow By System
One of the main functional requirements of our system is the work done by system. It includes simulation engine that produces match results from datas, outputs of functions, and user preferences to be written to the database instantly.

Description Of System Reports And Outputs
This requirement consists of listing the outputs of the simulation engine's or functions to the user and processing datas and producing results. It is required, because the flow of the game will be decided according to these outputs.

Control Of System Users
Since the game relies on the user choises and preferences, controlling the aforesaid user is important and required. Game can be started if user has an account in the database. If the user has a saved game account in database, he/she will be able to continue to his/her game, otherwise a new game database will be created.

Database Reading & Writing
By getting the data input, datas will be stored in database until the start of the work flow, which means the next function to be performed. Every information taken from the user is saved to the database instantly and is used then. This action makes database usage important.

2. Non-Functional Requirements

Backup & Recovery
Since we develop our project as a team and every member of the team contributes the codes, our system must be saved and secure.
Our project is being developed by using the most powerful revision control system, Git, on the website GitHub. This choice gives us to store our software securely, and restore the project at an unwanted situation.

Documentation
As a dependency of a team-work, each step of the project must be documented well. The methods that are used in software development process ease intelligibility. Another property of our documentation is “commit messages” stored in GitHub, which are written in every step of development.

Failure Management
A collaborated project may be vulnerable and any possible failure must be fixed. Failure management is handled with the help of the methods in our backup & recovery requirements.

Maintainability
Maintainability is one of the main non-functional requirements in our project, because we want to continue developing our project after the first release. Starting this point of view, we want to implement new features, technologies, and designs easily.

Open Source
Depending on maintainability requirement, we want our project to be developed by other developers who wants to. This requirement will make us to publish the codes to public in the end.

Performance
Performance is one of the key requirements in our project. It directly effects the quality of the project. That is the main reason for us to choose Python as the programming language and SQLite for the database management.

Platform Compatibility & Portability
This is decided to be a requirement, because of our request to play the game in different OS's. As a result of Python's compatibility and portability, users will be able to play the game by compiling the codes or getting the executable files after the release.

Usability
Usability is a non-functional requirement chosen for user-side of the game to give the user both easy and effective chance to play the game. Due to this requirement a simple interface and gameplay is designed for the game.