**Non-functional requirements:**

**\*\*These are better kept in this format**

The system must be robust and maintainable.

The system must be well documented, to allow another team to continue working on it if necessary.

The system must be easy to use for all users.

The system must be scalable.

The system must display the AUT logo on the interface at all times where possible.

**Infrastructure requirements**

As a client, I want the system to be set up in less than 3 hours so that It can be setup before the competition.

As a client, I want the network to be accessible to users through most mobile devices so that more users can make use of the application.

As a client, I want the network to handle over 500 users without decreasing performance so that no user gets frustrated trying to use the application.

**Application**

As a client, I want the application to handle requests from at least 500 users at one time without reducing performance so that a real-time sensation is preserved.

As a client, I want the application to be accessible to users through most mobile devices so that more users can make use of the application. **(compare to Infrastructure requirements)**

As a client, the application must be released on GPL or another open-source license so that it can be further expanded by future groups.

As a client, I want the application to provide secure connection to privilege users so their information is secure.

**Backend:**

As a client, I want the backend to be set up in less than 3 hours so that It can be setup before the competition. **(compare to Infrastructure requirements)**

As a client, I want the backend operational system to run on Linux so that no proprietary licence is needed.

As a client, I want the database to support concurrency so that real time feaure can be achieved.

**Database**

Unable to create all database user stories without overlapping with the other domains. The database diagrams provide the tables needed to build the application. The database should be created as features are developed.

As a client, I want that the database store:

* a list of teams in the competition.
* the answers for each team.
* the score for each team
* the number of passes for each team.
* timestamps for correctly answered questions.
* login credentials for privileged users

As a client, I want a team of a competition to be assigned to one marker only.

As client, I want to store the start time of each competition so we can run a countdown.

**Admin User Stories:**

As an admin, I want to add teams to the scoreboard system so that I can setup the competition.

**Marker User Stories:**

As a marker, I want to enter a correctly answered question so that the system can proceed to the next question.

As a marker, I want to enter an incorrectly answered question so that the database is updated with number of attempts.

As a marker, I want to enter a passed question so that the database is updated with number of passes remaining and the system can proceed to the next question.

As a marker, I want to be able to undo my previously input so that I can reverse any unwanted actions.

As a marker, I want to enter login credentials to login so that access to my account is secured.

As a marker, I want to be able to select the team to marker when I login to my account so I can choose between the teams assigned to me.

**Spectator User Stories:**

As a spectator, I want to see the scoreboard in a graphical format so that the scores are easier to interpret.

As a spectator, I want to constantly update the scoreboard so that the system functions in real time.

As a spectator, I want to select a particular team so that I can view their stats specifically.

As a spectator, I want to be able to search for a team so I can find and focus on that particular team.

As a spectator, I want to see a team’s stats when I select it.

As a spectator, I want to choose how the scoreboard is sorted so it is catering to my needs.

As a spectator, I want the scoreboard to sort by the highest by default.