Requirements Document

**1. Problem Statement**

**(i) Who Am I?**

The player is asked to think of a famous celebrity. A set of questions is then asked and based on the responses, the celebrity is guessed by the software.

**(ii) Tic-Tac Toe**

One of the players is allowed to login to the software so that he knows all the possible strategic moves to his opponent’s moves and hence will always win.

**2. Background Information**

**(a) Background Information about the problem, the domain, and targeted users.**

**(i) Who Am I?**

The idea of this game originates from an European version of it to guess celebrities. The algorithms that have been plied include a binary search from a decision tree that contain the database of celebrities. The aim is to find the celebrity in at most 20 questions.

The domain of the database can include 2^20=1,048,576 possibilities. A unique combination of the 20 answers gives the name of the celebrity. The questions are designed such that the combination of answers is unique.

It is a time-killer application that targets users of any age group for entertainment purpose.

**(ii) Tic Tac Toe**

This is a well-known game played between 2 players in a 3x3 grid. However,making only one player win always is the task of the application.

The domain of the game covers all the possible combinations of 0’s and x’s.

It is a game designed for users of all age groups.

**(b) Applications or systems those are similar to the planned work.**

There are existing games of Tic Tac Toe that are very common. There is an existing application “akinator genie” that predicts the celebrity using questions. It however, displays random questions rather than fixed questions (which we plan to use). It’s database is huge.

**(c) Limitations**

**(i) Who Am I?**

* The database required is huge and never-ending.

SOLUTION: To allow users to add a new celebrity if his/her name does not exist in the database.

* More than one celebrity has same set of answers for the questions asked.

SOLUTION: To design a questionnaire such that all celebrities have a unique combination of answers.

**(ii) Tic Tac Toe**

* If one of the solutions fails in making the player win.

SOLUTION: To extensively study as many cases as possible.

* If the grid size is increased from 3X3 to a higher number, the complexity of finding a strategic move increases.

SOLUTION: We stick to playing in a 3X3 grid.

**3. Stakeholders**

 A stakeholder in the architecture of a system is an individual, team, organization, or classes thereof, having an interest in the realization of the system.

The **baseline stakeholders** for both the games are:

**A) USERS:** In our project, the users are the players of the games. They require the services of the game to be fast, accurate and attractive in front design.

**B) DEVELOPERS:** The developers are stakeholders but their stake in the final requirements specification, or indeed in the system itself, is different from the users. They are more concerned in achieving the target, i.e. Correct answer to “Who Am I?” and Win-Win situation for the player in Tic-Tac-Toe.

**Other stakeholders:**

**C) FEEDBACK PROVIDING AND FREQUENT USERS:** These are the subset of the users that can actually find errors in strategic moves in Tic Tac Toe. Only frequent users can do this since they try new moves every time they play. They can provide with celebrities whose names are not present in the database and hence help in updating.

**4. Requirements**

**FUNCTIONAL REQUIREMENTS**

Following are the functional requirements for the games: (I.D.s are provided for priority)

**i) WHO AM I?**

* Attractive front end**(4)**

As a user, I would like the display page to be colorful and attractive.

* Accuracy**(1)**

I would like each of the celebrities I think of to be correctly identified. Only such a game will make itself interesting to play.

* Wide range of options**(2)**

The celebrities available in the game should be of a wide range.

**(The database should be huge and correct.)**

* Feedback Provision**(3)**

In case, a celebrity is not available in the database, I should be able to suggest the addition into the database.

**ii) TIC TAC TOE**

* Attractive front end**(2)**

As a user, I would like the display page to be colorful and attractive.

* WIN-WIN SITUATION**(1)**

The game should always allow me to win the game and not the computer or the other non-registered player.

**(This step is tedious itself and requires all cases from a user’s view to be considered for a win.)**

**NON-FUNCTIONAL REQUIREMENTS**

(e) Include use case diagrams

(f) In the Non-Functional Requirements subsection, please discuss performance and platform requirements such as response time, scalability, usability, security, etc. This can be either in a “user stories” format with detailed explanations or just a discussion for each requirement.