# 2) Requirements

Requirement can be defined as the singular documented physical and functional need that a particular product or service must be or perform. It is a raw need which requires external inputs or efforts to deliver a final product.

Requirement analysis is the process of analyzing, discovering, defining and documenting the requirements. This involves measuring of each and every functional, non-functional, technical and operational requirement. It is the discussion and formal agreement between the client and the development agency regarding what is expected of the product. This stage of waterfall model has been the important step in Software Development Life Cycle (SDLC) because this clearly states what will be in the scope and what is out of the scope. Here we have performed analysis and discovered the functional and non-functional requirements for the ***Cars Game***. The requirements identified are clearly stated and described in the following sections.

## 2.1) Functional Requirements

These requirements are the desired functionality that the client want us to build and be delivered to them. A functional requirement describes the interaction of system with its environment like what are the inputs to the system, what are the outputs from the system considering all the external and internal factors involved. The functional requirements for the ***Cars Game*** were analyzed and identified. They are described below in detail: -

ID: FR1

Title: Player sign up

Description: A new player should be able to sign up with the game in order to create personal profile or account for the game. The player must provide some information for the sign up as requested.

Rational: To acquire player credentials for login credentials

Dependencies: FR2

ID: FR2

Title: Player log in

Description: The administrator or a registered player both should be able to login to the game using their own login credentials. Appropriate message should be provided to them while logging in.

Rational: To maintain privacy and security of player accounts

Dependencies: FR1

ID: FR3

Title: Edit Player Profile

Description: A player should be able to edit his/her profile in the game. Appropriate message should be provided after the profile has been edited to the player.

Rational: To allow users to update their details for the profile in the game

Dependencies: FR2

ID: FR4

Title: View the highest score so far in the game with scorer name

Description: A player should be able to view the highest score scored so far and the name of the player who scored it.

Rational: To create a competitive environment in the game

Dependencies: FR2

ID: FR5

Title: View player’s personal high score

Description: A player should be allowed to view his/her personal best score so far in the game and compare with the highest score of the game.

Rational: To allow player to compare personal score with highest score of the game

Dependencies: FR2

ID: FR6

Title: Start the game

Description: A registered player should be able to start the game from the main dashboard. They should be navigated to the main game panel when he/she starts the game.

Rational: To navigate player to the main game panel.

Dependencies: FR2

ID: FR7

Title: Control the cars

Description: The game consists of 2 cars and 2 tracks respectively. A player should be allowed to control these two cars at the same time through separate keystrokes for each car in order to change the lane.

Rational: To allow player to control the cars in the game through keystrokes.

Dependencies: FR2, FR6

ID: FR8

Title: Collect Circles

Description: During the game runtime, a player should be able to collect circles on the track. Circles can be collected by controlling the cars to change the lanes for each track.

Rational: To

Dependencies: FR2, FR6, FR7

ID: FR9

Title: Avoid Squares

Description: While playing the game, a player should be able to avoid squares on the track. Squares can be avoided by controlling the car to change the lanes for each track.

Rational:

Dependencies: FR2, FR6, FR7

ID: FR10

Title: Pause Game

Description: A player playing the game should be able to pause the game. All the moving objects and the game time should be paused whenever a player pauses the game. The game can be paused using a ‘Pause’ button or any special keystroke specified by the game.

Rational:

Dependencies: FR2, FR6

ID: FR11

Title: Resume Game

Description: When a player pauses the game, the player should also be able to resume it. All the paused objects and the game time should continue just after the player resumes the game. The game can be resumed using ‘Resume’ button.

Rational:

Dependencies: FR2, FR6, FR10

ID: FR12

Title: Exit to dashboard

Description: After a player pauses the game, the player should be allowed to exit the game and navigate to the dashboard. He/she can exit to the main menu by clicking ‘Exit’ button displayed after the game has been paused.

Rational:

Dependencies: FR2, FR6, FR10

ID: FR13

Title: Log Out

Description: The administrator or a player both should be provided with the functionality to log out from the game. They can simply do this by pressing ‘Log Out’ button from the dashboard or main menu.

Rational:

Dependencies: FR2

ID: FR14

Title: Reset Scores

Description: The administrator of the game should be allowed to reset all the scores scored by the players in the game. The admin can do this by clicking ‘Reset’ button from the admin dashboard.

Rational:

Dependencies: FR2

## 2.2) Non-functional Requirements

Non- functional requirements are the untold parts of the project which are not communicated but really understood as a global standard. They are always supporting the functional requirements. They can also be defined as constraints on a system like performance speed, security and many more which can be optimized to a certain level of extent. The non-functional requirements for the project were identified. They are described below in detail: -

# 3) Prioritization