# SOFTWARE ENGINEERING PROBLEM SPECIFICATION TABLE

|  |  |
| --- | --- |
| CUSTOMER | APO II COURSE |
| USER | Players of PIPEMANIA |
| FUNCTIONAL REQUIREMENTS | FR0: Main Menu  FR1: Starting menu.  FR2: Game creation.  FR3: Editing the board.  FR4: Simulation.  FR5: Scoreboard. |
| CONTEXT OF THE PROBLEM | It is necessary to develop a software based on a video game called PipeMania, which consists of placing pipes per console and simulating the flow of water. The objective is to transport the water correctly from the source to the drain, for this there are 3 types of pipes: vertical (| |), horizontal (=) and 90° (o) joints.  Then the system is responsible for using double-linked lists to simulate the flow of water within a plane of 8X8 squares, to later calculate a score relating the hits and the execution time, and finish showing the user a table of scores corresponding to per player. |
| NON-FUNCTIONAL REQUIREMENTS | NFR0: Validation of system inputs.  NFR1: Presentation of the interface by console.  NFR2: System scalability. |

## Functional requirements analysis table

| Name or identifier | FR0: Main Menu | | |
| --- | --- | --- | --- |
| Summary | When starting the system must be presented in console in main menu for the user, which contains 3 options. The first is the option to start the game, the second is to see score and the third is to finish. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| option | Int | Option between [1,2,3] |
| Result or postcondition | The system validates the input and executes the corresponding option. | | |
| Outputs | Output Name | Data type | Format |
| msgValidation | String | msg |

| Name or identifier | FR1: Starting menu | | |
| --- | --- | --- | --- |
| Summary | Once the game is created, the system must present a menu of 3 options (Edit board, Simulate water flow, Exit). Once the option is entered, the system validates the data and executes the option. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| option | Int | Option between [1,2,3] |
| Result or postcondition | The system validates the input and executes the option. | | |
| Outputs | Output Name | Data type | Format |
| msgValidation | String | .msg |

| Name or identifier | FR2: Starting | | |
| --- | --- | --- | --- |
| Summary | The system must allow the user to start a game from the main menu, entering his nickname and then creating the 8X8 board with the water source (F) and drainage (D) located randomly. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| nickname | String | Invalid nickname |
| Result or postcondition | The system generates the board to the user and presents a new menu to play the game. | | |
| Outputs | Output Name | Data type | Format |
| board | String | Board in game |
| gameMenu | String | Menu with options |

| Name or identifier | RF3: Editing the dashboard | | |
| --- | --- | --- | --- |
| Summary | During the game the system presents the option to insert pipes their corresponding validated positions on the board, if the pipe is correct and its position is free, then the board is updated with the new modification, | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| pipetype | Int | [1,2,3,4] |
| Row | Int | [1-8] |
| column |  | [1-8] |
| Result or postcondition | The system validates the inputs and executes the change by updating the board with the new pipeline. | | |
| Outputs | Output Name | Data type | Format |
| board | String | Board updated |

| Name or identifier | RF4: Simulation | | |
| --- | --- | --- | --- |
| Summary | The system must simulate the flow of water through the pipes from the source to the drain if the pipes connect correctly, it is verified that there are no interruptions during the journey and that the circular pipes only connect 2 pipes in 90°.  Then, the user is validated if the answer is correct and the game is finished, storing the time spent (sg) and the number of pipes used to continue with the calculation of the score.  If the answer is not correct, the game continues until the user finds the solution or leaves. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| N/A | N/A | N/A |
| Result or postcondition | The system validates whether the user journey is correct and proceeds with the assignment of the score. If the system with the game is not correct. | | |
| Outputs | Output Name | Data type | Format |
| msgValidation | String | msg |

| Name or identifier | RF5: Score Table | | |
| --- | --- | --- | --- |
| Summary | The system must have a scoring system represented by a binary tree. This score table must be distributed from highest to lowest per player and show all the scores found in it. The user will see the ordered score table in the console. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| N/A | N/A | N/A |
| Result or postcondition | The score table will receive a player which will be saved to know your score at any time. | | |
| Outputs | Output Name | Data type | Format |
| scoreBoard | String | msg |