| Client | Computer distribution company |
| --- | --- |
| User | Potential customers and employees of the company |
| Functional requirements | * R1. The program must allow the user to start a build that will be an array of components. * R2. The program must be able to suggest a build automatically based on 2 components chosen by the user, one belonging to the lowest level (CPU) and the other to the highest level (GPU).   + Build the cheapest build. * R3. The program should allow the user to make his build manually by filtering the compatible parts as he chooses each type of component. * R4.The choice of components and visualization of results should be displayed through a graphical interface. |
| Context | A computer distribution company needs a program that automatically shows builds and recommends components based on the user's choices. |

| Name | **R1: Start a Build** | | |
| --- | --- | --- | --- |
| Summary | The system should initialize the variables that will contain each component | | |
| Inputs | **Input Name** | **Date type** | **Condition** |
|  |  |  |
| General Activities | The graph should been initialized and don’t be empty | | |
| Result | Fill the variables with R2 or R3 | | |
| Outputs | **Output Name** | **Date type** | **Condition** |
|  |  |  |

| Name | **R2: Make a build with two restrictions** | | |
| --- | --- | --- | --- |
| Summary | The system must create a Build from two components selected by the user, one from the lowest level (CPU) and the other to the highest level (GPU) | | |
| Inputs | **Input Name** | **Date type** | **Condition** |
| start | node | The component is included in the graph |
| final | node |
| General Activities | Execute Dijkstra between both nodes, with the condition that there is no more than 1 component of the same type. | | |
| Result | Create the list of components of the shortest path (by price) | | |
| Output | **Output name** | **Date type** | **Condition** |
| parts | ArrayList<Node> | if there is a path |

| Name | **R3: Filter components while making the build** | | |
| --- | --- | --- | --- |
| Summary | The system should allow the user to choose the pieces that he wants to be part of his computer and based on that filter the next components in the hierarchy | | |
| Inputs | **Input Name** | **Date type** | **Condition** |
| cpuComponent | node | The components should exist in the graph |
| motherBoardComponent | node |
| ramComponent | node |
| gpuComponent | node |
| storageComponent | node |
| General Activities | Run DFS from each selection that the user makes to filter the next components and ensure that they're compatible | | |
| Result | Return the list of components that the user selected for each type | | |
| Inputs | **Output name** | **Date type** | **Condition** |
| parts | ArrayList<Node> |  |

| Name | **R4:** Graphic interface | | |
| --- | --- | --- | --- |
| Summary | The system must allow the user to choose components and display the results of his choices through a graphical interface. | | |
| Inputs | **Input Name** | **Date type** | **Condition** |
| clickAction | T |  |
| General Activities | Detect requested action and execute the corresponding action. | | |
| Result | Display actions through the interface | | |
| Inputs | **Output name** | **Date type** | **Condition** |
| answer | T |  |