|  |  |
| --- | --- |
| **Functional Requirements** | **Non-functional Requirements** |
| Provide a graphical user interface to the user | The graphical user interface should be nice to look at and easy to use. |
| Generate a list of patients in the Critical care unit. | Limit access to certain individuals depending on their user privilege. |
| Filter patients based on the category e.g., Body mass index | N/A |
| Flag patients who need to see a dietitian depending on the data provided | The ability to run on multiple different platforms (windows, Mac and Linux) |
| Review individual patient data and allow the user to edit it. | Good performance when doing more complex tasks |
| Generate reports on individual patients e.g., bar and pie charts | N/A |
| Create and delete user accounts |  |

**Functional requirements:**

A graphical user interface must be provided to the user because a graphical user interface is easier to use than a command line interface therefore reducing the training costs.

A list of patients must be generated to allow for the patient data to be visualized using the graphical user interface.

Allow users to filter the patients based on various categories such as body mass index (BMI) to facilitate targeted intervention.

Flag patients who require dietitian consultation based on specific criteria, enhancing resource allocation and patient care prioritization

Enable users to review and modify individual patient data to ensure accuracy and completeness.

Produce comprehensive reports on individual patient data using graphical representation like bar and pie charts for better understanding and decision-making.

Provide functionality to securely create and delete user accounts to manage user access and permissions effectively.

**Non -Functional Requirements**

The graphical user interface should be visually appealing and user-friendly to enhance user experiences and usability.

Access control mechanisms must be implemented to restrict system access based on user roles and privileges, ensuring data security and confidentiality.

The system should be compatible with various operating systems to ensure accessibility and flexibility for users regardless of their platform preferences.

The system should demonstrate efficient performance, particularly when handling more complex operations or large datasets , to ensure responsiveness and usability .