## **Part 2 –**

## **Functional Requirements:**

### **System Requirements:**

* The user selects the different factors displayed in the graph. If no criteria are selected, a default option will be displayed.
* The user can select two factors for each graph and no more.
* The system will be able to transfer the new data uploaded by the user to the database.
* The system presents the user with several options for the type of graph and the selection of 2 criteria.
* After the user chooses the type of graph and the types of criteria, the system generates a graph according to the user's choices.
* The system will support a variable length of information fields about patients.

### **Model Requirements:**

* Training a model based on different algorithms and storing it in the database.
* The model will be able to identify relationships between the different data.
* The model will be able to determine whether a specific patient will return for rehospitalization.

### **Database Requirements:**

* To synthesise information as close to the source as possible so the hospital will not encounter problems related to the file format when using the system.
* The database holds several trained models.
* A database that holds information about patients.

### **Non-Functional Requirements:**

### **Design Requirements:**

* A straightforward and user-friendly interface allows new users without a technological background to operate the application themselves from the first use.
* Easy design to prevent user confusion.

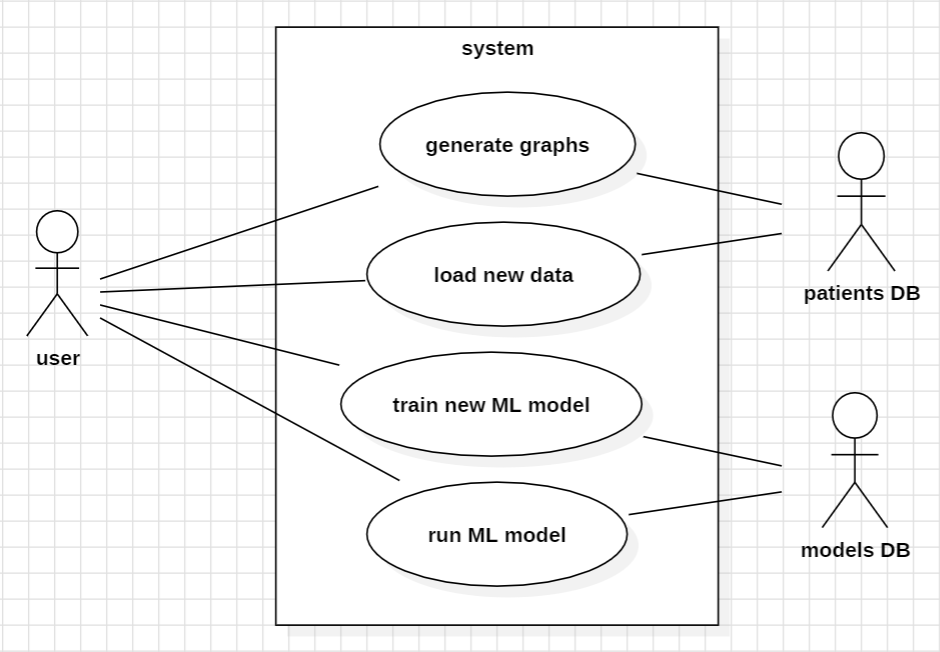
### **Performance Requirements:**

* The system can create a requested graph according to the selected criteria in a minimal time (to be defined later).
* We will define a specific level of accuracy that we will require from our model for finding relationships between the data.

### **Operational Requirements:**

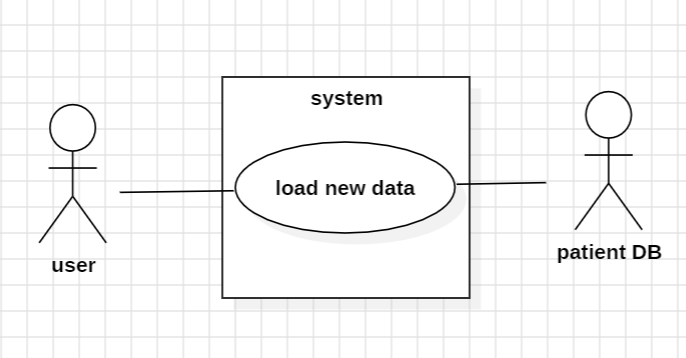
* The system can interface with a CSV file and read data from it according to the format provided by the hospital.

**Use Case Diagram - Initial Functional Analysis**

****

**Use cases**

**Loading New Patient Information**

****

**Main Flow:**

* The user selects the "Upload File" option from the menu**.**
* The system prompts users to upload a CSV file containing hospitalized patient data.
* The system uploads the data to the hospitalized patient database.
* The system checks if the new data entered already exists in the system.
* The system records the new data in the patient database.

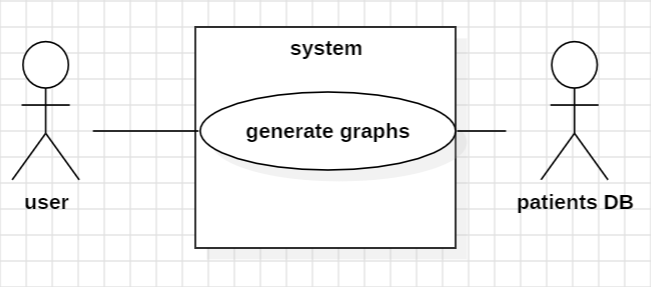
**Secondary Flow A:**

* This secondary flow diagram follows step 2 in the main flow.
* The file is not uploaded or not opened correctly by the system.
* The system sends the user a message about the error and asks the user to try again or upload a different file.
* If the new upload is successful, we proceed to step 3 in the main flow. Otherwise, we return to step 1**.**

**Secondary Flow B:**

* This secondary flow diagram follows step 4 in the main flow.
* The system checks if the new data entered already exists in the system.
* The system identifies that the data already exists and will not update the patient database.

**Generate Graph**

****

## **Main Flow:**

1. The user selects the "Create Graph" option from the menu.
2. A menu opens, displaying a list of available graph types.
3. The user selects the desired graph type.
4. A menu opens, displaying the data fields available in the patient database.
5. The user selects two fields.
6. The system extracts the desired information from the patient database.
7. The system generates a graph based on the user's selections**.**

## **Secondary Flow A:**

This secondary flow diagram follows step 5 in the main flow.

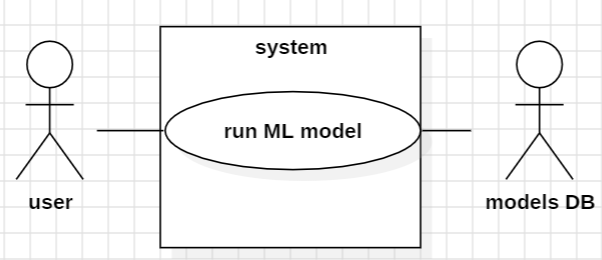
* The user selects too few or too many fields.
* An error message is sent to the user, and another attempt is made. We return to step 4 in the main flow**.**

## **Secondary Flow B:**

This secondary flow diagram follows step 5 in the main flow.

* There is no patient information in the database.
* An error message is sent to the user, and a message is sent to the user to upload a data file.
* We proceed to the new data upload scenario**.**

**Run Machine Learning Algorithm**

****

## **Main Flow:**

1. The user selects running machine learning algorithm.
2. A list of trained models stored in the database opens.
3. The user selects a trained model.
4. The user is requested to load a new patients file in CSV format.
5. The user loads a new patients file.
6. The algorithm runs on the new data.
7. The system shows the user the result of whether the patient will return for rehospitalization.

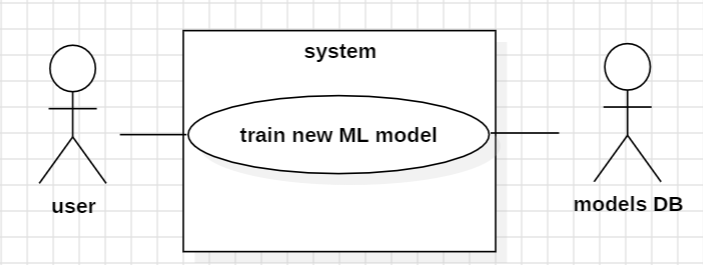
## **Secondary Flow A:**

* This secondary flow diagram follows step 2 in the main flow.
* There are no trained models stored in the database.
* We proceed to the model training scenario.

**Secondary flow B:**

* This secondary flow diagram follows step 4 in the main flow.
* User load invalid format file.
* Systems displays an error message and request the user to try and load a different file.
* Getting back to step 5 of the main flow.

**Train New Model**

****

## **Main Flow:**

* The user selects the "Train Model" option from the menu.
* A list of possible algorithms for training the model is displayed.
* The user selects an algorithm to train the model.
* The system trains the model according to the selected algorithm.
* The system saves the trained model in the model database**.**

## **Secondary Flow:**

This secondary flow diagram follows step 4 in the main flow.

* Model training fails.
* The system sends the user the reason for the failure and asks for another attempt - we return to step 3.