## **Part 2 –**

## **Functional Requirements:**

### **System Requirements:**

* The user selects the factors displayed in the graph. If no criteria is selected, a default option will be displayed.
* The user can select one factor 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 criteria.
* After the user chooses the type 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.
* The system will give the user the option to generate a graph according to the filter selected.

### **Model Requirements:**

* Training a model based on different algorithms and storing it in the database.
* The model will be able to make predictions due to the data uploaded by the user.
* The model will be able to determine: A. the Hospitalist doctor's level of certification. B. The number of days passed between the first hospitalization and the second one.

### **Database Requirements:**

* The database holds several trained models.
* A database that holds information about hospitalization and re hospitalization.

### **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.

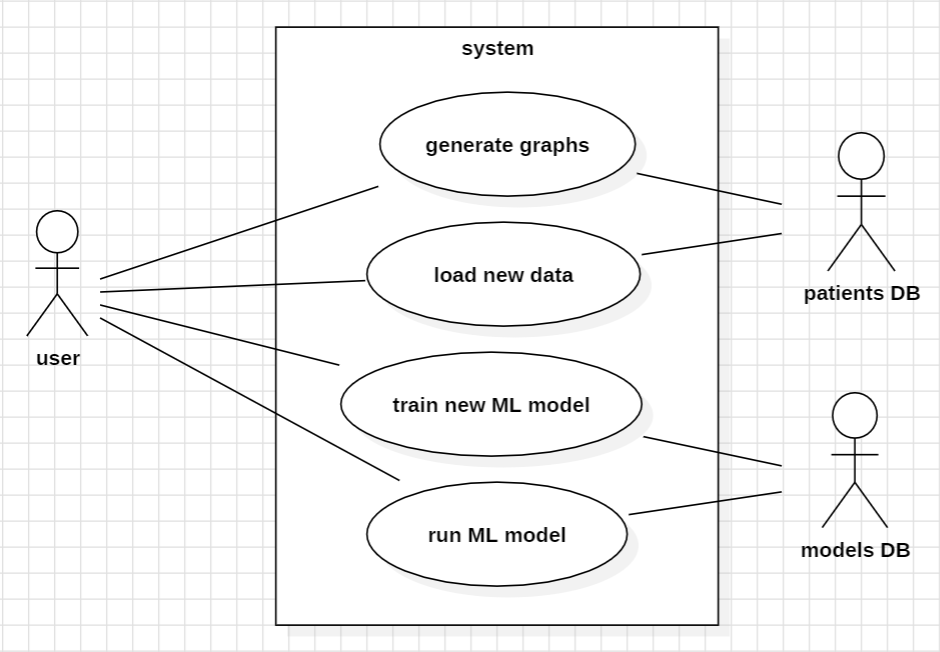
### **Performance Requirements:**

* 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 prediction making. (to be defined later).

### **Operational Requirements:**

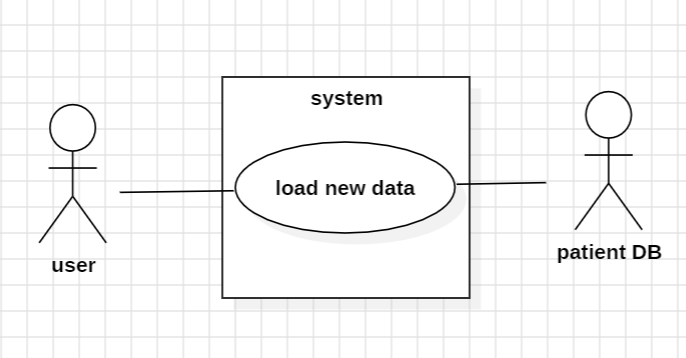
* The system can interface with a CSV file and read data from it according to the format provided by the hospital.
* The system can adjust itself to the changes made in the file – in cases of adding or removing a field. (nice to have).

**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.

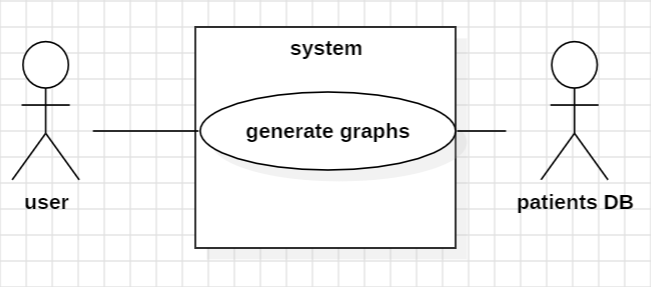
**Secondary Flow A:**

* 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 the data fields available in the patient database.
3. The user selects one field.
4. The system extracts the desired information from the patient database.
5. The system generates a graph based on the user's selections**.**

## **Secondary Flow A:**

This secondary flow diagram follows step 3 in the main flow.

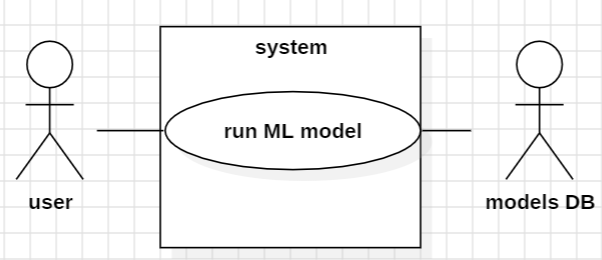
* The user selects too few or too many fields.
* An error message is sent to the user, and another attempt is made. We return to step 4 in the main flow**.**

## **Secondary Flow B:**

This secondary flow diagram follows step 3 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 run machine learning algorithm.
2. A list of trained models stored in the database opens.
3. The user selects a trained model.
4. The user needs to select the prediction type.
5. The user needs to upload a new CSV file that contains all the data, for the ML to run on.
6. The algorithm runs on the new data according to the prediction type.

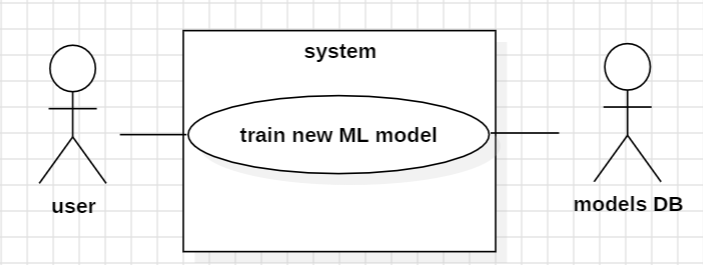
## **Secondary Flow A:**

* 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 5 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:**

1. The user selects the "Train Model" option from the menu.
2. A list of possible algorithms for training the model is displayed.
3. The user selects an algorithm to train the model.
4. The system trains the model according to the selected algorithm.
5. 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.

**Generate Filtered Graph**

תמונה שמכילה תרשים, קו, טקסט, עלילה

התיאור נוצר באופן אוטומטי

## **Main Flow:**

1. The user selects the "Create Filtered Graph" option from the menu.
2. A menu opens, displaying the data fields available in the patient database.
3. The user selects one field.
4. The user sets a filter according to the field he chose.
5. The system extracts the filtered information from the hospitalization database.
6. A menu opens, displaying the data fields available in the patient database.
7. The user selects one field as a parameter shown in graph.
8. The system generates a graph based on the user's selections**.**

## **Secondary Flow A:**

This secondary flow diagram follows step 3 and 6 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 or 7 accordingly in the main flow**.**

## **Secondary Flow B:**

This secondary flow diagram follows step 3 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**.**

## **Secondary Flow C:**

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

* The system fails to filter the data according to the chosen filter.
* An error massage with the failure reason will be sent to the user.
* In the case of data corruption, the system will ask the user to upload data again or to check its validity.
* The system will make another attempt to filter file.
* Returning to main flow , step 6.