# Classes:

* Note
* NoteManager
* MainForm
* Vital
* ValidationHelper

# Class Note

Class to create the note. The object Note.

## Properties

* NoteId
  + Int
* PatientName
  + String
* PatientDOB
  + Datetime
* PatientProblems
  + List string
* NoteContent
  + String

# Class NoteManager

Class to manage the notes. Write and read the notes from the file.

## Properties

* List with the notes
  + List for object Note
* filePath
  + String

## Methods

* LoadNote
  + Load the notes from the file
  + Display all the content
    - Using foreach
* SaveNote
  + Save/Write the note to the file
* UpdateNote
  + Update the content of a specific note
    - Using the specific index
* AddNote
  + Add a new note
    - Managing the UniqueId
* DeleteNote
  + Delete a specific note
    - Using the specific index
* ListNotes
  + List with all notes

# Class MainForm

Class with all the EventHandlers of the form.

# Class Vital

Class to manage the Vitals. Could use Interface here.

## Properties

* VitalValue
  + The value of the Blood Pressure and Heart Rate
* VitalType
  + Blood Pressure or Heart Rate
* VitalUnit
  + mmHg for BP
  + bpm for HR
* IsLow
  + Boolean
* IsHigh
  + Boolean

## Methods

* CheckBP
  + Check if the BP is low, high or normal
* CheckHR
  + Check if the HR is low, high or normal
* AddHighOrLow
  + Add the High or Low to the string
* AddUnit
  + Add the unit to the string
* ListVitals
  + List all the Vitals after the methods above are done

# Class ValidationHelper

Class with methods to handle all the required validations.

## Methods

* IsUniqueId
  + Manage the unique NoteId
* ValidateInput
  + Vaildate all the inputs as requested