

hospit.Application Class Reference

Public Member Functions

```
def __init__(self, master)
def appoint(self)
def add_appointment(self)
def update(self)
def chercher_db(self)
def modifier(self)
def supprimer(self)
def voirRendezVous(self)
def func(self)
def Information(self)
```

Public Attributes

master

self.master: le master de l' objet courant on va le passer le master qu' on la pris comme parametre dans le constructeur **init**

hautFrame

self.hautFrame est un cadre inclut dans le fenetre master qu'on la met en haut de cette fenetre dont la longueur est 1200 et la largeur est 150 et avec couleur lightsteelblue

framebienv

self.framebienv est un cadre inclut dans le fenetre master positionné avec abscisse x=163 et ordonnée 0 dont la longueur est 1200 et la largeur est 150 et avec couleur 'midnightblue'

cadreInfo

self.cadreInfo est un cadre inclut dans le fenetre master qu'on la met en bas de cette fenetre par rapport à hautFrame qui est en haut avec une largeur de 1200 et longueur de 620 et dont la couleur est 'lightsteelblue'

image

self.image est une variable dans laquel on va mettre une image de format png 'logo.png' ouvert en utilisant 'Image.open' importé de la bibliotheque Pillow qu'on le doit l'installé pour mettre des images

photo

self.photo est une variable dans laquel on va mettre l'image ouvert self.image en utilisant ImageTk.PhotoImage importé de la bibliotheque Pillow

canvas

self.canvas est une zone rectangulaire destiné a contenir des images et autres figures et cette variable self.canvas est crée dans le cadre hautFrame pour mettre dedans notre image 'logo.png' self.canvas.create_image permet de creer ou mettre notre objet image "logo.png " dans le canvas

saisir

self.saisir est un bouton appelé 'Saisir un rendez-vous' mit dans le cadre 'self.hautFrame' qui renvoie vers la fenetre 'appoint(self)' qui est une fonction, saisir a une largeur de 30 ,longueur de 2, police de caractère 'arial 10 bold' et couleur background:steelblue , couleur foreground :blanche

chercher

self.chercher est un bouton appelé 'chercher et modifier rendez-vous' mit dans le cadre 'self.hautFrame' qui renvoie vers la fenetre 'update(self)' qui est une fonction, chercher a une largeur de 37 ,longueur de 2, police de caractère 'arial 10 bold' et couleur background:steelblue , couleur foreground :blanche

voir

self.voir est un bouton appelé 'Voir Les rendez-vous' mit dans le cadre 'self.hautFrame' qui renvoie vers la fenetre 'voirRendezVous(self)' qui est une fonction, voir a une largeur de 30 ,longueur de 2, police de caractère 'arial 10 bold' et couleur background:steelblue , couleur foreground :blanche

info

self.info est un bouton appelé 'Plus d'information' mit dans le cadre 'self.hautFrame' qui renvoie vers la fenetre 'Information(self)' qui est une fonction, info a une largeur de 34 ,longueur de 2, police de caractère 'arial 10 bold' et couleur background:steelblue , couleur foreground :blanche

bienvenue
self.bienvenue est une étiquette(Label) appelé 'LOGICIEL DE GESTION DES RENDEZ-VOUS DES PATIENTS' mit dans le cadre 'self.framebienv', bienvenue a police de caractère 'arial 22 bold' et couleur background:midnightblue , couleur foreground :blanche
result
self.result va prendre le resultat de l'execution de la requete "SELECT * FROM appointments" avec c.execute(requet)
rows
self.rows va prendre tout les lignes de self.result par fetchall()
framevide
self.framevide: Si self.rows retourne une liste vide ,pas de resultat de la requete alors on crée une cadre self.framevide dans le cadre self.vadreInfo , avec largeur 1200 ,longueur 510 et couleur background 'midnightblue'
bdVide
self.bdVide: est une étiquette dont le nom est 'Aucun patient n'est saisi ,veuiller saisir un patient..' écrit dans le cadre self.framevide dans le cas ou la requete retourne resultat vide donc la liste self.rows est vide
texte
self.texte: dans le cas ou self.rows retourne une resultat alors on crée une zone de texte de variable self.texte dans le cadre self.cadreInfo de largeur 140 et longueur 30 positionné avec abscisse de 25 et ordonnée 25,
fen1
fen1 est la fenetre dans laquel on va saisir les rendez-vous des patients et en affichant le journal des saisis
left
self.left c'est le cadre à gauche de fen1 dans laquel ov saisir les informations d'un patient et cette cadre prend toute la longueur de fen1 et prend un largeur de 800px
right
self.right c'est le cadre à droite de fen1 dans laquel dans lequel on va afficher le journal des patients saisis , cette cadre prend toute la longueur de fen1 et prend un largeur de 400px
heading
etiquette 'label' represente le titre du formulaire saisie pour le patient
nom
etiquette 'label' du nom du patient saisie
prenom
etiquette 'label' du prenom du patient saisie
age
etiquette 'label' de l'age du patient saisie
phone
etiquette 'label' du numero telephone du patient saisie
genre
etiquette 'label' du genre du patient saisie
location
etiquette 'label' de location du patient saisie
temps
etiquette 'label' de temps du rendez-vous du patient saisie
nom_ent
entrez nom du patient
prenom_ent
entrez prenom du patient
age_ent
entrez l'age du patient
phone_ent
entrez le numero telephone du patient

variable
self.variable est une variable de type String qui prend 2 valeurs pour le menu d' option de genre soit 'masculin' ou 'feminin'
genre_ent
self.genre_ent est une menu d'option pour entrer le genre du patient et cette menu prend 2 valeurs 'masculin' ou 'feminin'
variable2
self.variable2 est une variable de type String qui varie selon le choix de l'utilisateur dans une liste d'options pour la location et cette liste prend les differentes villes en Mauritanie
location_ent
self.location_ent est une menu de liste d'option permet le choix de location parmi plusieurs villes de Mauritanie
variable3
self.variable3 est une variable de type String qui varie selon le choix de l'utilisateur dans une liste d'options pour le temps prévu du rendez-vous et cette liste prend des durées de temps bien precis
temps_ent
self.temps_ent est une menu de liste d'option permet le choix de temps prévu pour le rendez-vous parmi plusieurs temps bien précisé
submit
self.submit c'est la bouton qui permet d'ajouter un rendez-vous
retourner
self.retourner c'est la bouton qui nous permet de retourner au fenetre d'acceuil
log
self.log c'est une étiquette qui donne le titre 'journal' qui affiche le journal du patient saisie
box
self.box c'est une zone de texte dans laquel on va afficher le journal du patient saisie
id
self.id va prendre toute les valeurs de la colonne des IDs des patients
new
rendre les ids par ordre croissant
final_id
val1
obtenir le nom du patient saisie
val2
obtenir le prenom du patient saisie
val3
obtenir l'age du patient saisie
val4
obtenir le genre du patient par le choix de l'un des options
val5
obtenir le Location du patient par le choix de l'un des options
val6
obtenir le numero du telephone du patient saisie
val7
obtenir le temp du rendez-vous du patient par le choix de l'un des options
fen2
self.fen2 est une fenetre sera ouvert quand on clique sur le bouton 'chercher et modifier un rendez-vous' dans le fenetre d'acceuil
CadreFen2
self.CadreFen2 est un cadre qui prend l'ecran plein de la fenetre self.fen2 en lui ajoutant une couleur background 'lightsteelblue'

head self.heading est une étiquette qui represente le titre de la fenetre qui est "chercher un rendez-vous et le modifier"
name self.name est une étiquette 'label' a pour texte "entrer le nom du patient:"
entr_name self.entr_name est une entrée qui permet de saisir le nom du patient afin de le saisir
search self.search est un bouton qui permet chercher un patient en envoyant la commande de la recherche vers la fonction chercher_db qui retourne les informations du patient
retourner2 self.retourner2 est un bouton qui permet de retourner au fenetre precendent c'est à dire la fenetre d'acceuil
input_name self.input_name permet de recupère le nom du patient saisie par l'utilisateur
Prenom self.Prenom permet de recupère le prenom du patient cherché d'après la base de donnée
temps_prevu self.temps_prevu permet de recupère le temps prévu de rendez-vous du patient cherché d'après la base de donnée
lab_nom self.lab_nom est une étiquette qui a pour texte 'Nom du patient'
lab_prenom self.lab_prenom est une étiquette qui a pour texte 'Prenom du patient'
lab_age self.lab_age est une étiquette qui a pour texte 'Age'
lab_genre self.lab_genre est une étiquette qui a pour texte 'genre'
lab_location self.lab_location est une étiquette qui a pour texte 'Location'
lab_phone self.lab_phone est une étiquette qui a pour texte 'Numero Tel'
lab_temps self.lab_temps est une étiquette qui a pour texte 'temps du rendez-vous'
entr_nom self.entr_nom est une entrée dans laquelle on insert le nom du patient cherché par l'utilisateur d'après la base de donnée
entr_prenom self.entr_prenom est une entrée dans laquelle on insert le prenom du patient cherché par l'utilisteur d'après la base de donnée
entr_age self.entr_age est une entrée dans laquelle on insert l'age du patient cherché par l'utilisteur d'après la base de donnée
entr_genre self.entr_genre est une entrée dans laquelle on insert le genre du patient cherché par l'utilisteur d'après la base de donnée
entr_location self.entr_location est une entrée dans laquelle on insert la location du patient cherché par l'utilisteur d'après la base de donnée
entr_phone self.entr_phone est une entrée dans laquelle on insert le numéro du telephone du patient cherché par l'utilisteur d'après la base de donnée
entr_temps

self.entr_temps est une entrée dans laquelle on insert le temps prévu du rendez-vous du patient cherché par l'utilisateur d'après la base de donnée
update self.update est un bouton permet de modifier les informations d'un patient à l'aide de la fonction 'modifier'
delete self.delete est un bouton permet de supprimer le rendez-vous d'un patient à l'aide de la fonction 'supprimer'
var1 self.var1 est une variable prend le nom du patient affiché d'après la recherche de rendez-vous de ce patient afin de permettre à l'utilisateur de le modifier
var2 self.var2 est une variable prend le prénom du patient affiché d'après la recherche de rendez-vous de ce patient afin de permettre à l'utilisateur de le modifier
var3 self.var3 est une variable prend l'age du patient affiché d'après la recherche de rendez-vous de ce patient afin de permettre à l'utilisateur de le modifier
var4 self.var4 est une variable prend le genre du patient affiché d'après la recherche de rendez-vous de ce patient afin de permettre à l'utilisateur de le modifier
var5 self.var5 est une variable prend la location du patient affiché d'après la recherche de rendez-vous de ce patient afin de permettre à l'utilisateur de le modifier
var6 self.var6 est une variable prend le numéro du telephone du patient affiché d'après la recherche de rendez-vous de ce patient afin de permettre à l'utilisateur de le modifier
var7 self.var7 est une variable prend le temps prévu du rendez-vous du patient affiché d'après la recherche de rendez-vous de ce patient afin de permettre à l'utilisateur de le modifier
run self.run prend la requete de modification et le faire executer ,dès qu'il affiche une alerte de succès d'execution alors la modification se réalise dans la base de donnée
x self.x est l'indice que les listes 'number' et 'patient' prend pour configurer le nom et numéro du patient
fen3 self.fen3 est une fenetre sera ouvert quand on clique sur le bouton 'voir les rendez-vous' dans le fenetre d'acceuil
cadrefen3 self.cadrefen3 est un cadre qui prend l'ecran plein de la fenetre 'self.fen3' dont la couleur du background est 'lightsteelblue'
heading3 self.heading3 est une etiquette qui a pour texte "Rendez-vous" qui represente le titre de la fenetre
retourner3 self.retourner3 est un Boutton pour revenir au fenetre precedent
change self.change est un Boutton pour faire changer le patient en donnant le patient suivant
n self.n est une étiquette 'Label' indiquant le numero du rendez-vous dont le texte est vide qu'on le configurera et on parlera plutard avec la fonction 'func'
pname self.pname est une étiquette 'label' indiquant le nom du patient dont le texte est vide qu'on le configurera et on parlera plutard avec la fonction 'func'
fen4 self.fen4 est une fenetre ouvrit en cliquant sur le bouton 'plus d'information' dans la fenetre d'acceuil ,elle contient des informations sur ce logiciel
cadrefen4 self.cadrefen4 est un cadre qui prend toute la fenetre et lui ajoute une couleur background 'lightsteelblue'
heading4 self.heading4 est une étiquette 'label' qui donne un titre pour la fenetre "Informations sur ce Logiciel"

retourner4

self.retourner4 est un Boutton pour revenir au fenetre precedent

Inform1

self.Inform1 c'est une étiquette qui donne la 1ère information

Inform2

self.Inform2 c'est une étiquette qui donne la 2ème information

Inform3

self.Inform3 c'est une étiquette qui donne la 3ème information

Inform4

self.Inform4 c'est une étiquette qui donne la 4ème information

Inform5

self.Inform5 c'est une étiquette qui donne la 5ème information

Inform6

self.Inform6 c'est une étiquette qui donne la 6ème information

Inform7

self.Inform7 c'est une étiquette qui donne la 7ème information

Detailed Description

Application c'est la classe dans laquelle on va faire notre projet en créant un constructeur et des différents attributs et méthodes et à la fin on crée un objet qui fait appel à cette classe en passant comme paramètre 'root' qui ouvre la fenêtre principale et ce paramètre correspond à celui déclaré dans le constructeur

Constructor & Destructor Documentation

◆ __init__()

```
def hospit.Application.__init__( self,  
                                master  
                                )
```

Dans ce constructeur on prend comme paramètre d'entrée self et master, self indique l'objet courant, master indique la fenêtre principale (accueil) quand on le remplace par 'root' en créant un objet b=Application(root) et dans cette fenêtre on va faire notre travail en ajoutant cadres, images, textes etc..

Member Function Documentation

◆ add_appointment()

```
def hospit.Application.add_appointment ( self )
```

add_appointment est une fonction qui prend les valeurs saisie par l'utilisateur ,c'est à dire les informations du patient, si l'un des champs n'est pas encore remplie ,l'application va afficher une alerte qui demande à remplir tout les champs sinon les informations du patients seront saisies dans la table 'appointments' de notre base de donnée et dès que les données sont saisie il va nous affiche une alerte de la succès de l'insertion et dans le journal on va récupérer les données que nous venont de saisir en affichant aussi le nombre des patients saisis

◆ appoint()

```
def hospit.Application.appoint ( self )
```

'appoint' est une fonction dont on va crée une fenetre fen1 dans laquel on va saisir les rendez-vous des patients avec son nom,prenom,son age ,location et son date prevu pour 1 et dans le fenetre de cette fonction on va lire les journaux des patients saisis .

◆ chercher_db()

```
def hospit.Application.chercher_db ( self )
```

chercher_db est une fonction qui permet de chercher les informations du rendez-vous d'un patient selon le nom de ce patient et elle recupère ces informations de notre base de donnée et elle les affiche sur le meme fenetre ou on l'a chercher et avec cette affichage elle nous propose 2 boutons l'un permet la modification des informations du rendez-vous et l'autre permet la suppression du rendez-vous

◆ func()

```
def hospit.Application.func ( self )
```

func est une fonction pour parler et modifier le texte des étiquettes self.n et self.pname qui represente le numero du rendez-vous et le nom du patient et pour ce faire on va déclarer deux listes vides 'numbrer' et 'patients' et par une requete on prend les colonnes de la table 'appointments' et on remplit les listes 'number' et patients' par les colonnes qui contient nom et numero du patient et puis on fait la configuration pour que l'utilisateur puisse voir un rendez-vous en cliquant sur le bouton 'le patient suivant'

◆ Information()

```
def hospit.Application.Information ( self )
```

Information est une fonction avec laquel on crée une fenetre 'self.fen4' dans laquel on donne des informations sur ce logiciel concernant necessairement l'objectif de ce type de logiciel pour que l'utilisateur puisse bien comprendre le but,l'utilisation et savoir qui aura le besoin de l'utiliser

◆ modifier()

```
def hospit.Application.modifier ( self )
```

modifier est une fonction qui permet de modifier les informations d'un patient cherché par l'utilisateur et si ces informations sont modifiées dans la base de donnée ,elle va nous affiche une alerte qui indique la succès du modification

◆ supprimer()

```
def hospit.Application.supprimer ( self )
```

supprimer est une fonction qui permet à l'utilisateur de supprimer un rendez-vous d'un patient cherché avec toute ses informations et ca sera une suppression definitive de la base de donnée et dès que la suppression se réalise ,il s'affiche une alerte indiquant la succès de la suppression

◆ update()

```
def hospit.Application.update ( self )
```

update c'est une fonction qui permet de créer une fenetre self.fen2 et il permet aussi de chercher dans cette fenetre un rendez-vous d'un patient par son nom et le modifier ou supprimer si on a besoin de le faire

◆ voirRendezVous()

```
def hospit.Application.voirRendezVous ( self )
```

voirRendezVous est une fonction permet de crée une fenetre self.fen3 et dans cette fenetre elle va afficher par ordre le patient qui doit entrer chez le docteur en cliquant sur le bouton 'patient suivant' et cette fonction va utilisé une autre fonction funct(self) pour configurer l'affichage du nom et numero du patient par ordre

The documentation for this class was generated from the following file:

- C:/Users/t440/Desktop/Projet Python 2019/Code/hospit.py