// Fermer & Afficher interface
   GtkWidget *.

accueil = lookup_widget (objet, "accueil")   ← plafe.
gtk_widget_hide (accueil);
         destroy

Gestion_Entity = create_gestion_Entity();
gtk_widget_show ("   ");

_____

// Recuperation de contenu
1) déclarer les ≠ pointeurs .GtkWidget *

2) ~~Des~~ Récuperer les @ des widgets ds les ≠ pointeurs
   id = lookup_widget (objet, "EntryId");
   nom = lookup_widget (objet, "EntryNom");
   Type = lookup_widget (objet, "ComboType");
      ⋮

3) - Récuperer les textes des widgets et copier ds des chaines
   strcpy (e.ID, gtk_entry_get_text(GTK_ENTRY(id));
   strcpy(e.Type, gtk_combo_box_get active_text(GTK_CONBO_B
   .d.jour = gtk_spin_button_get_value_as_int (GTK_SPIN_BUTTON(
      ↳ si jour declaré ds la struct come entier
                                              x = ahoi
      sinon
              sprintf (e.d.jour, "%d", ...);

   if (sexe == 1) . strcpy (e.sexe, "Male");
   else if (sexe == 2) . strcpy (e.sexe, "Female");
   ↙ variable globale: int sexe = 0;   /   int choix[] = {0,0,0};
                        ✓ à modifier avec le signal Toggl
          if (gtk_toggle_button_get_active(GTK_RADIO_BUTTON(togg
                  ... = 1; /   choix[2] = 1;

// Mise à jour du contenu des objets graphiques

// Déclaration + Récuperation des @ widgets

Combobox

`gtk_combo_box_set_active ( GTK_COMBO_BOX ( comboType), i);` // GTK widget @

Spin button

`gtk_spin_button_set_value ( Spin1, jour);`

Label

`gtk_label_set_text ( GTK_LABEL (l_LabelOutput), "Code incorrect");` jour = 15;

RadioButton

`if (strcmp (U.role, "Admin") == 0)`
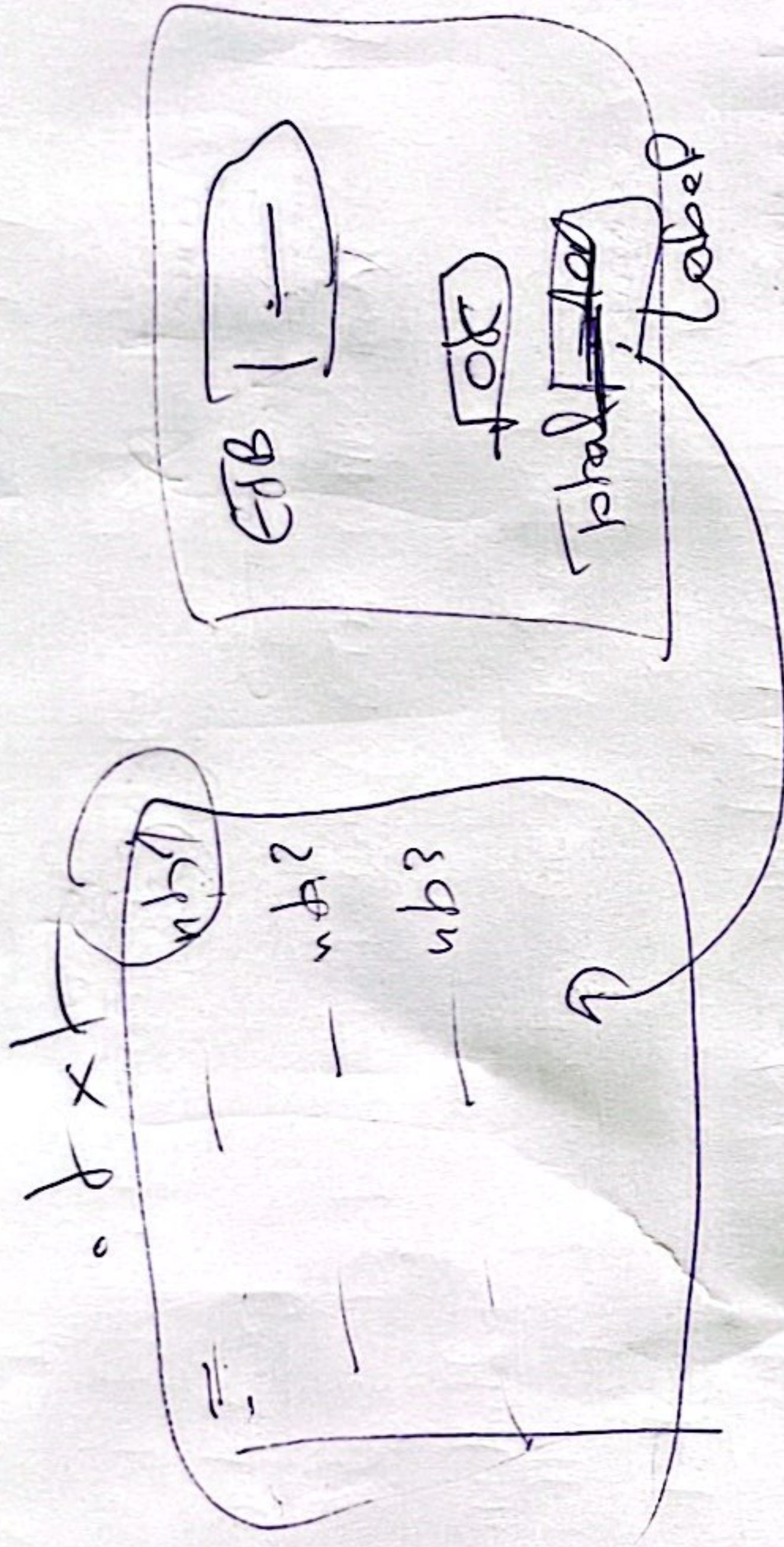`gtk_toggle_button_set_active ( GTK_RADIO_BUTTON ( Radio), TRUE);`
`else if (` _____
`else if (` _____

Case à cocher

`int T[3] = {0,1,1};`
`if (T[0] == 1)`
`gtk_toggle_button_set_active ( GTK_TOGGLE_BUTTON (toggleBout1));`
`if ( T[1] )` _____

X = atoi (str); // convertir une chaîne en entier

Sprintf (ch, "%d", i); // convertir un entier en chaîne.

Résumé S11 - Projet 3B/20

// Fermer & Afficher Interface

GtkWidget *
accueil = lookup_widget (objet, "accueil");

gtk_widget_hide (accueil);
→ destroy

Gestion_Entity = Create_gestion_Entity();

gtk_widget_show (" ");

// Récupération de contenu

1) Déclarer les # pointeurs .GtkWidget

2) Récupérer les @ des widgets ds les # pointeurs
   id = lookup_widget (objet, "EntryId");
   nom = lookup_widget (objet, "EntryNom");
   Type = lookup_widget (objet, "ComboType");
   .....

3) Récupérer les textes des widgets et les ranger ds des chaînes

strcpy(e.ID, gtk_entry_get_text (GTKENTRY(id)));

strcpy(e.Type, gtk_combo_box_get_active_text(GTK_COMBO_BOX(Type)));

e.d.jour = gtk_spin_button_get_value_as_int (GTK_SPIN_BUTTON(j));
   → Si jour déclaré ds la struct comme entier
   sinon
   sprintf (e.d.jour, "%d", ...);

if (sexe == 1). strcpy (b.sexe, "Male");
else if (sexe == 2) strcpy (e.sexe, "Femall");
→ variable globale int sexe =0;   int choix [J = {0,0,0}; ...
   à modifier avec le signal Toggled
if (gtk_toggle_button_get_active (GTK_RADIO_BUTTON(togglebutton))

x = choi (str);