

Dès mon arrivé à l'E2C, mon tuteur de stage Patrick m'a donné une tâche, celle de travailler sur l'application PRISM, dans un stade déjà très avancé pour espérer la lancer prochainement.

Lors de l'entré d'un stagiaire à l'E2C on lui fais passer des tests pour évaluer son niveau global sur différent domaine, ce sont les « Test de positionnement de l'école de la seconde chance ». L'e2c La garde accueille chaque semaine environ 7 à 10 stagiaires, les tests de positionnement sont imprimés et font entre 8 et 10 pages, c'est une impression conséquente sur l'année. Afin de réduire le cout monétaire ET écologique Patrick, responsable du pole informatique de l'UPV a donc décider de les numériser et d'en faire une Appli. PRISM.

C'est dans ce contexte que l'application s'est créer et que je suis amené à travailler dessus depuis le premier septembre 2025.

PRISM est une application développée sous Power Apps pour plusieurs raisons :

- La première ; la simplicité d'apprentissage du langages low code power FX
- La deuxième ; le lien avec l'e2c. En effet l'e2c à pour but de former des employés dans tout les domaines possible et imaginable dont le développement, développer en low code permet de prendre des stagiaires de notre propre structure pour les introduire au développement d'application.

Avant d'être nommé PRISM l'application s'appelait PosE, pour positionnement Electronic.

P
R
I
S
M



(menu accueil de PRISM, Le prénom se charge automatiquement avec le nom de compte e2c-var ou upv.org de l'utilisateur)

A mon arrivé en tant que développeur à l'E2C l'application était déjà bien avancé, les questions apparaissent les menues et écrans sont fait et les réponses sont déjà sélectionnable.

Cependant rien n'était stocké ! Il était déjà prévu de stocker les données dans un Excel dans un SharePoint mais cette partie-là n'avais pas encore été abordé. Ma première mission fût donc de permettre la sauvegarde de

donné dans cet Excel. D'abord les Comptes/infos, ensuite les réponses sélectionnées.

```
// Initialisation des variables pour stocker les réponses progressives
Set(ReponsesUserQCM; ""); 
Set(ReponsesUserQCU; ""); 
// On met une variable pour stocker la réponse en cour.
Set(QuestionEnCoursId; LookUp(Suivi; Compte = User().Email).EnCours)
```

Dans la propriété OnVisible de l'écran Accueil j'ai stocké les réponses localement dans un premier temps.

```
//Alimentation des champs
If(
  IsBlank(LookUp(Suivi; Compte = User().Email));
  Patch(
    Suivi;
    Defaults(Suivi);
    {
      Compte: User().Email;
      Prénom: First(Split(User().FullName; " ")).Value;
      Nom: Last(Split(User().FullName; " ")).Value;
      Site: LookUp(accountCreation; Mail = User().Email; Bureau.Value);
      Secteur: SecteurChoisi;
      Complet: false;
      Formateur: LookUp(accountCreation; Mail = User().Email; Responsable.Value);
      'Début Test': Now()
    }
  );
  Patch(
    Suivi;
    LookUp(Suivi; Compte = User().Email);
    {
      Prénom: First(Split(User().FullName; " ")).Value;
      Nom: Last(Split(User().FullName; " ")).Value;
      Site: LookUp(accountCreation; Mail = User().Email; Bureau.Value);
      Secteur: SecteurChoisi;
      Complet: false;
      Formateur: LookUp(accountCreation; Mail = User().Email; Responsable.Value);
      'Début Test': Now()
    }
  )
);;
```

Dans la propriété OnSelect du bouton commencer le quizz je stock toutes les infos de bases de nos stagiaires vers notre Excel PRISM_Suivi

Le compte, le prénom, le nom, le site, le secteur, si le test est fini (non par défaut), le secteur choisi pour ce quizz (BTP – SAT – TRA etc...), le formateur référent du stagiaire, la date de début du test.

La séparation Isblank permet de soit créer la ligne sur le Excel soit la mettre à jour.

Ensuite il a fallu stocké nos réponses !

```

// 1. Mettre à jour les variables locales séparément
If(
    TypeQuestion = "QCM";
    Set(
        ReponsesUserQCM;
        ReponsesUserQCM & If(!IsBlank(ReponsesUserQCM); " // "; "") &
        LookUp(ListeQuestionsQCM; IntituleQuestion = IntituleQuestion).Id & " : " &
        If(
            Len(Concat(Filter(ListeReponsesQCM; Repondu = "OUI"); ContenuTexte & " / ")) > 0;
            Left(
                Concat(Filter(ListeReponsesQCM; Repondu = "OUI"); ContenuTexte & " / ");
                Len(Concat(Filter(ListeReponsesQCM; Repondu = "OUI"); ContenuTexte & " / ")) - 3
            );
            ""
        )
    );
);

If(
    TypeQuestion = "QCU";
    Set(
        ReponsesUserQCU;
        ReponsesUserQCU & If(!IsBlank(ReponsesUserQCU); " // "; "") &
        LookUp(ListeQuestionsQCU; IntituleQuestion = IntituleQuestion).Id & " : " & ReponduQCU
    );
);

// 2. Patch les colonnes correspondantes en fusionnant avec les anciennes valeurs
If(
    TypeQuestion = "QCM";
    Patch(
        Suivi;
        LookUp(Suivi; Compte = User().Email);
        {
            ReponsesQCM:
            If(
                !IsBlank(ReponsesUserQCM);
                If(
                    !IsBlank(LookUp(Suivi; Compte = User().Email).ReponsesQCM);
                    LookUp(Suivi; Compte = User().Email).ReponsesQCM & " // " & ReponsesUserQCM;
                    ReponsesUserQCM
                );
                LookUp(Suivi; Compte = User().Email).ReponsesQCM
            )
        }
    );
);

If(
    TypeQuestion = "QCU";
    Patch(
        Suivi;
        LookUp(Suivi; Compte = User().Email);
        {
            ReponsesQCU:
            If(
                !IsBlank(ReponsesUserQCU);
                If(
                    !IsBlank(LookUp(Suivi; Compte = User().Email).ReponsesQCU);
                    LookUp(Suivi; Compte = User().Email).ReponsesQCU & " // " & ReponsesUserQCU;
                    ReponsesUserQCU
                );
                LookUp(Suivi; Compte = User().Email).ReponsesQCU
            )
        }
    );
);

```

```

// 3. Mettre à jour l'ID de la question en cours
Patch(
  Suivi;
  LookUp(Suivi; Compte = User().Email);
{
  EnCours: If(
    TypeQuestion = "QCM";
    LookUp(ListeQuestionsQCM; IntituleQuestion = IntituleQuestion).Id;
    LookUp(ListeQuestionsQCU; IntituleQuestion = IntituleQuestion).Id
  )
}
)

```

Toutes ces lignes de code sont dans la propriété OnSelect du bouton valider réponse de l'écran quizz.

Plus loin dans le quizz on initialise la valeur 'fin test' sur le Excel qui si existe et est supérieur à la date de début quizz met la valeur 'complet' sur True pour dire que le stagiaire à fini son test

```

Patch(
  Suivi;
  LookUp(Suivi; Compte = User().Email);
{
  'Fin Test': Now()
}
);

```

Après cela mon tuteur m'a demandé de permettre de reprendre le quizz si on quitte l'application avant la fin du quizz

```

  If(
    !IsBlank(LookUp(Suivi; Compte = User().Email).EnCours);
    Navigate(EcranReprise; ScreenTransition.None)
)

```

j'ai créer un bouton invisible avec timer , a la fin du timer si la valeur Encours du Excel (qui permet de connaitre la question en cours) est vide alors rien ne se passe, dans le cas contraire on reprend là on en était !

A ce stade l'appli est « fonctionnelle », on peut répondre au quizz on stock les données. Cependant certains médias ne sont pas appelés, certaines questions nécessitent une image pour répondre et cette image ne s'affiche pas. A l'heure actuel 06/11/2025 nous n'avons pas encore trouvé la solution. Cela s'explique car beaucoup de personne différente ont travaillé sur cette application et certains médias sont stockés sur server d'autre sur internet directement. L'application fonctionne avec une application sœur

nommé ECHO, sur laquelle j'ai travaillé dès sa création qui permet d'évaluer le stagiaire par son formateur référent !

