

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2023
Épreuve E5 - Conception et développement d'applications (option SLAM)	
ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle (recto)	

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 2								
Nom, prénom : Damien Monchaty		N° candidat : 02148833006								
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>	Date : 25/04/2023								
Organisation support de la réalisation professionnelle Dans le cadre d'une demande d'un site d'actualité de sport automobile FormulaNews, j'ai dû concevoir une application mobile diffusant les résultats de toutes les courses de Formule 1 depuis 1950. Elle permettra de plus d'avoir des informations sur le championnat en cours en termes de pilotes, constructeurs & classement										
Intitulé de la réalisation professionnelle FormulaNews										
Période de réalisation : Du 24.01.2022 au 07.04.2023 Lieu : Domicile – Candidat Libre Modalité : <input checked="" type="checkbox"/> Seul(e) <input type="checkbox"/> En équipe										
Compétences travaillées <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input type="checkbox"/> Gérer les données										
Conditions de réalisation¹ (ressources fournies, résultats attendus) Ressources fournies : Le site d'actualité FormulaNews aimerait avoir une application mobile diffusant plusieurs actualités concernant la formule 1 pour gagner en audience. Il n'existait jusqu'à lors aucun code au préalable. Résultat attendu : Quelques fonctionnalités tels que : <ul style="list-style-type: none"> - Information sur la prochaine course & dernière course - Consultation d'article – Mise en place d'un blog - Consultation des résultats de toutes les courses de Formule 1 depuis 1950 : - Classement Pilotes & Constructeurs - Toutes les courses et leurs résultats - Tous les pilotes et leurs résultats - Une utilisation facile & intuitive de l'application 										
Description des ressources documentaires, matérielles et logicielles utilisées² <table border="0"> <tr> <td>- Visual Paradigm Online</td> <td>- GitHub</td> </tr> <tr> <td>- Microsoft Office</td> <td>- Trello</td> </tr> <tr> <td>- Microsoft Visual Studio Community 2022</td> <td>- Visual Studio Code</td> </tr> <tr> <td>- SQLite</td> <td></td> </tr> </table>			- Visual Paradigm Online	- GitHub	- Microsoft Office	- Trello	- Microsoft Visual Studio Community 2022	- Visual Studio Code	- SQLite	
- Visual Paradigm Online	- GitHub									
- Microsoft Office	- Trello									
- Microsoft Visual Studio Community 2022	- Visual Studio Code									
- SQLite										
Modalités d'accès aux productions³ et à leur documentation⁴ La commission aura la possibilité d'accéder à l'ensemble de la solution en la téléchargeant à l'url suivante : https://damienmonchaty-btssio.com/spe2 La commission aura la possibilité d'accéder au cahier des charges en téléchargeant la solution à l'url suivante : https://damienmonchaty-btssio.com/spe2 La commission aura la possibilité d'accéder au manuel technique en téléchargeant la solution à l'url suivante : https://damienmonchaty-btssio.com/spe2 La commission aura la possibilité d'accéder à la documentation technique en téléchargeant la solution à l'url suivante : https://damienmonchaty-btssio.com/spe2										

¹ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

² Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

³ Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁴ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

Épreuve E5 - Conception et développement d'applications (option SLAM)

ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)**Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs**

Etant Candidat libre et seul sur ce projet, voici toutes les étapes de conception du projet FormulaNews et sa documentation.

- Github – FormulaNews-Web : <https://github.com/DamienMonchaty/FormulaNews-Web>
- Github : – FormulaNews-Mobile : <https://github.com/DamienMonchaty/FormulaNews>
- Trello : <https://trello.com/b/Y39APC9A/formulanews-web-app-mobile>
- Url – FormulaNews-Web : <https://formulanews.azurewebsites.net/>

- Etape 1 : Mise en place **Mode Projet**
 - Mise en place de la méthodologie **SCRUM**
 - Création de User-Stories via l'outil en ligne **Trello**
 - Mise en place d'un répertoire **GitHub**

- Etape 2 : **Analyse du cahier des charges selon le besoin du client**
 - Expression du besoin
 - Analyse de la demande
 - Analyse du **MPD - Modèle Physique de Données**
 - Conception de Diagramme **UML - Unified Modelling Language**

- Etape 3 : Conception du projet **FormulaNews Mobile en C# et Xamarin Forms**
 - Respect du patron architectural **MVVM – Model View ViewModel**
 - Conception Interface graphique **XAML**
 - Utilisation de **divers packages NuGet**
 - Mise en place d'**Entity Framework Core – ORM Object Relationnal Mapping**
 - Mise en place d'**Entity Framework SQLite**

- Etape 3 : Conception du projet **FormulaNews Web en Angular NodeJs**
 - Mise en place de **Composants Web**
 - Conception Interface graphique **HTML5, CSS3**

- Etape 4 : Ecriture des tests UI avec **Xamarin.UITest**
 - Ecriture de tests UI des interfaces graphiques de l'application mobile **FormulaNews**

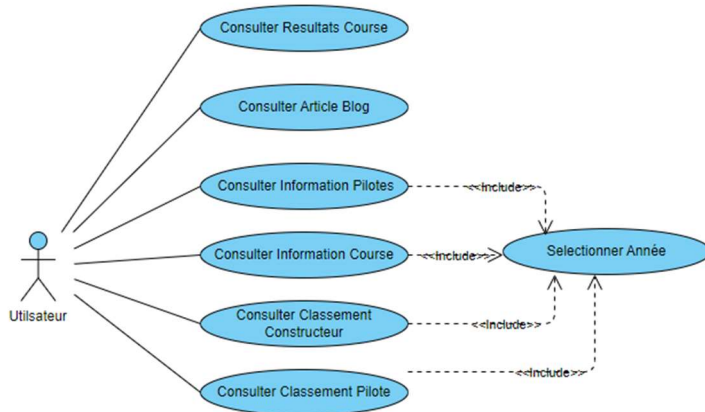
- Etape 5 : Mise en place de la documentation technique du projet **FormulaNews**
 - Génération de la documentation technique avec **Doxygen**

- Etape 6 : Mise en production de l'application Mobile Android **FormulaNews**
 - Packaging & Versioning de l'application
 - Protection de l'application
 - Compilation, Archive & distribution

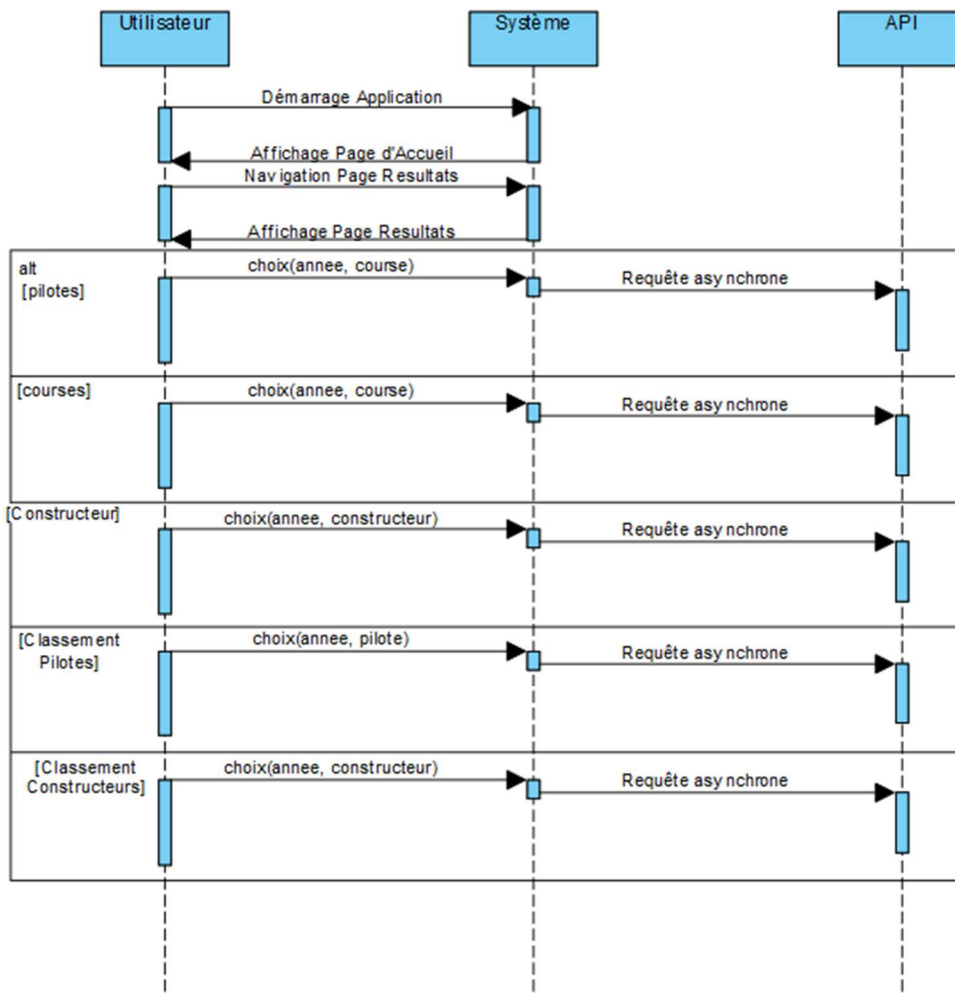
- Etape 7 : Mise en production de l'application Web **FormulaNews**
 - Mise en place d'un **Serveur NodeJs Cloud Azure** en liant son compte **GitHub**

➤ Plusieurs schéma-diagrammes UML ont été réalisées :

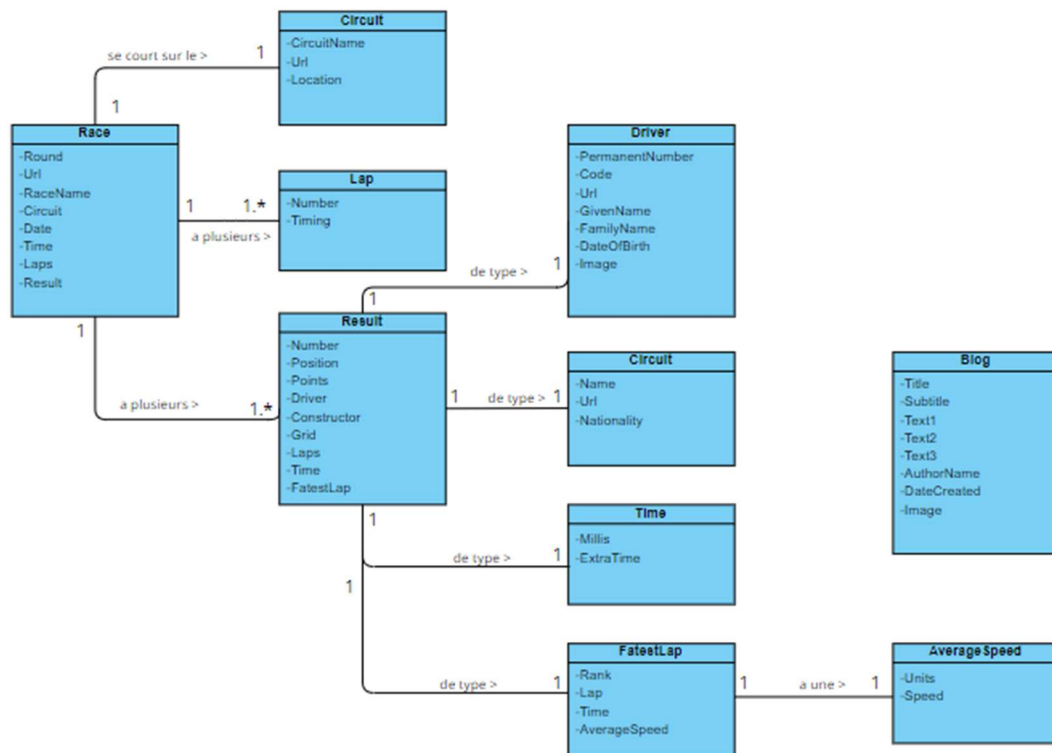
- Diagramme de cas d'utilisation



- Diagramme de séquence



- Diagramme de classes



➤ Gestion des données :

Pour gérer des données en **Xamarin Forms**, j'ai mis en place une base de données locale **SQLite**

Voici les étapes :

- Pour utiliser SQLite dans XF, il est nécessaire d'installer le paquet **NuGet sqlite-net-pcl**.
- Ecrire un code spécifique pour chaque plateforme IOS & Android

```

public interface ISQLiteDb
{
    SQLiteAsyncConnection GetConnection();
}

```

- Cette interface expose une méthode appelée **GetConnection()** qui sera implémentée dans chaque projet spécifique à la plate-forme et renverra la chaîne de connexion appropriée.

```

using Environment = System.Environment;
[assembly: Dependency(typeof(SQLiteDb))]
namespace FOne_MobileApp.Droid.Persistence
{
    public class SQLiteDb : ISQLiteDb
    {
        public SQLiteAsyncConnection GetConnection()
        {
            var documentPath = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
            var path = Path.Combine(documentPath, "MySQLite.db");
            return new SQLiteAsyncConnection(path);
        }
    }
}

```

- La classe BlogDao qui permet de récupérer des informations en base de données

```
public class BlogDao : IDao<Blog>
{
    private SQLiteAsyncConnection _connection;

    public BlogDao(ISQLiteDb db)
    {
        _connection = db.GetConnection();
    }

    public async Task<Blog> GetFirst()
    {
        return await _connection.Table<Blog>().FirstAsync();
    }

    public async Task<IEnumerable<Blog>> GetListAsync()
    {
        return await _connection.Table<Blog>().ToListAsync();
    }
}
```

- Pour insérer des données au démarrage de l'application, j'ai mis en place dans le projet **Android** une méthode **CreateDatabase()** dans la classe **MainActivity.cs**

```
[Activity(Label = "FormulaNews", Icon = "@drawable/logoicon", Theme = "@style/splashscreen", MainLauncher = true,
public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity
{
    SQLiteConnection dataBase;

    protected override void OnCreate(Bundle savedInstanceState)
    {
        base.OnCreate(savedInstanceState);
        this.CreateDatabase();
        Xamarin.Essentials.Platform.Init(this, savedInstanceState);
        global::Xamarin.Forms.Forms.Init(this, savedInstanceState);
        LoadApplication(new App());
    }

    public override void OnRequestPermissionsResult(int requestCode, string[] permissions, [GeneratedEnum] Android
    {
        Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

        base.OnRequestPermissionsResult(requestCode, permissions, grantResults);
    }

    private void CreateDatabase()
    {
        dataBase = this.GetConnection();
        dataBase.CreateTable<Blog>();

        dataBase.Query<Blog>("DELETE From Blog");
    }
}
```

➤ Ecriture des tests UI avec **Xamarin.UITest**

Xamarin.UITest est un framework de test C# utilisant **NUnit** pour les tests d'acceptation de l'interface utilisateur sur les applications iOS et Android.

Les tests interagissent avec l'interface utilisateur comme le ferait un utilisateur - en saisissant du texte, en appuyant sur des boutons et des gestes - tels que des balayages.

Test	Durée	C.	Me...
▲ FOne_MobileApp.UITest (3)	1.5 min		
▲ FOne_MobileApp.UITest (3)	1.5 min		
▲ Tests(Android) (3)	1.5 min		
BlogsDisplayed	33.5 s		
BlogsDisplayedFromHomeView	23.5 s		
ResultsDisplayed	31 s		

- La classe **Tests.cs**

```
[TestFixture(Platform.Android)]
public class Tests
{
    IApp app;
    Platform platform;

    public Tests(Platform platform)
    {
        this.platform = platform;
    }

    [SetUp]
    public void BeforeEachTest()
    {
        app = AppInitializer.StartApp(platform);
        app.Repl();
    }

    [Test]
    public void BlogIsDisplayed()
    {
        app.ScrollDown();
        app.WaitForElement(c => c.Marked("frame_blogDetails"));
        app.Tap(c => c.Marked("frame_blogDetails"));
        app.Screenshot("On " + this.GetType().Name);
    }

    [Test]
    public void BlogIsDisplayedFromHomeView()...

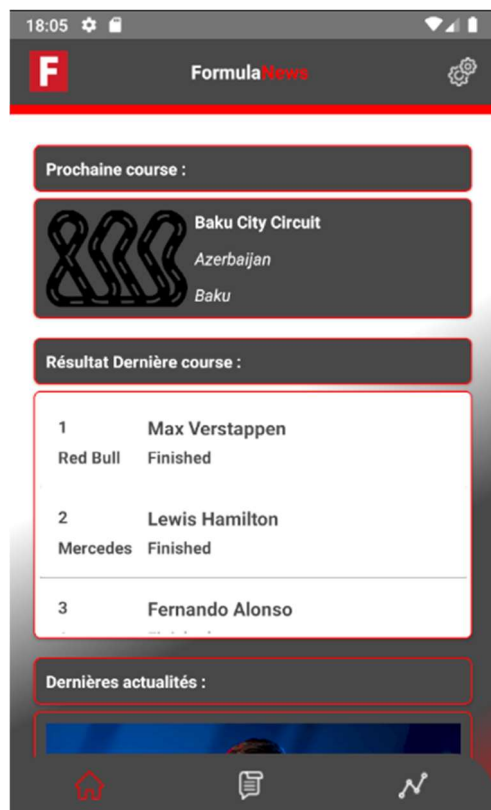
    [Test]
    public void ResultIsDisplayed()...
```

➤ Productions réalisées :

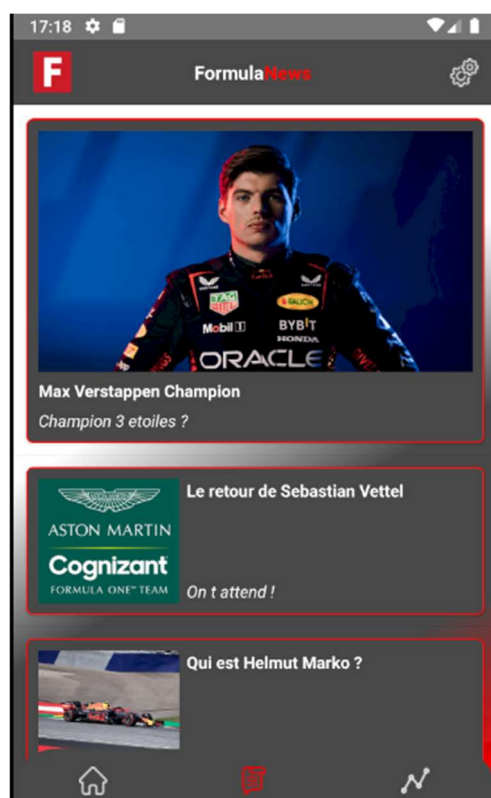
Les vues ont été conçues en **XAML - Extensible Markup Language**.

Il s'agit d'un langage de balisage déclaratif utilisé par les développeurs pour définir les interfaces utilisateur des applications **WPF & Xamarin Forms**.

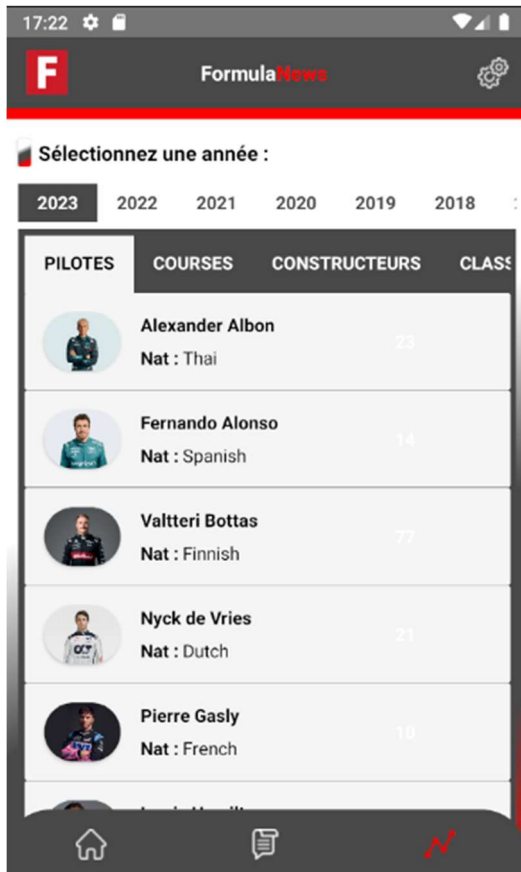
▪ Home



▪ Blog - Blog Détails



■ Resultats Année 2023 - Pilote Détails



Alexander Albon

Position : 10 **Alexander Albon**
57 tours Williams
 Finished **Temps : +1:29.774**

Position : 19 **Alexander Albon**
27 tours Williams
 Brakes **Temps :**

Position : 19 **Alexander Albon**
6 tours Williams
 Accident **Temps :**