

Todolist

Présentation du projet

Objectif	2
Déroulé	3
Spécifications fonctionnelles	4
Fonctionnalités de base	4
Fonctionnalités avancées	5
Exigences fonctionnelles et techniques	5
Exigences ergonomiques	6
Packages recommandés	7
Modalités de réalisation	8
Critères d'évaluation	8

Objectif

Votre projet consiste à concevoir et à développer une application mobile cross-platforms avec Flutter et le langage Dart en respectant les consignes fournies progressivement à chaque étape de TD.

L'application proposera des fonctionnalités de type *"Todo List"* : création et gestion de tâches.

Déroulé

La développement de l'application se déroule en **2 phases** :

- **Phase 1** : réalisation des **fonctionnalités de base**, en suivant les consignes fournies dans les supports de TP,
- **Phase 2** : sélection et réalisation en autonomie, de **fonctionnalités avancées** parmi une liste de fonctionnalités proposées.

Déroulé donné à titre indicatif.

Phase 1 :

1. **Étape 01** : initialisation du projet
4/6/2024 (séance de 2h),
2. **Étape 02** : modélisation des données
10/6/2024 (séance de 4h),
3. **Étape 03** : génération de données et affichage dynamique
10/6/2024 (séance de 4h),
4. **Étape 04** : navigation entre écrans (*Master / Details*) + formulaire d'ajout de tâche
10/6/2024 (séance de 4h),
5. **Étape 05** : mise en place du store de données *Provider*,
11/6/2024 (séance de 2h),
6. **Étape 06** : mise en place de l'API REST (*SupaBase*) et interactions + variables d'environnement,
12/6/2024 (séance de 4h),
7. **Étape 07** : personnalisation *UI / UX*,
12/6/2024 (séance de 4h) + 14/6/2024 (séance de 4h),
8. **Étape 08** : opérations de type CRUD via l'API REST,
14/6/2024 (séance de 4h),

Phase 2 :

9. **Étape 09** : sélection et réalisation d'au moins 1 fonctionnalité avancée,
14/6/2024 (séance de 4h),
10. **Étape 10** : finalisation des fonctionnalités avancées + rendu.
19/6/2024 (séance de 4h).

Spécifications fonctionnelles

Fonctionnalités de base

L'application permettra à un utilisateur de :

- **créer et gérer une liste de tâches** (ajout / modification / suppression),
- **naviguer à travers les différents écrans de l'application,**
- **associer un ou plusieurs tag(s) à une tâche,**
- **attribution un niveau de priorité à une tâche,**
- **interagir avec une *API REST* afin de :**
 - **s'authentifier,**
 - **sauvegarder ses données** dans une base de données distante,
 - **effectuer des opérations de type *CRUD*** sur ses données.

Fonctionnalités avancées

Après avoir réalisé les fonctionnalités de base, vous choisirez au moins 1 fonctionnalité avancée parmi cette liste :

- Gestion du **mode offline** (détection de l'absence de réseau + base de données locale + synchronisation avec l'API) + gestion des permissions,
- Création de **tâches intégrant un contenu graphique** (image capturée avec la caméra et/ou issue de la bibliothèque de médias) + gestion des permissions,
- **Géolocalisation de tâches** (soit, des tâches à réaliser à un emplacement GPS déterminé, ex : domicile, bureau...).
- **Association de tâche à un contact** (interaction avec le carnet de contact natif *iOS / Android*),
- **Intégration d'une tâche au calendrier** (interaction avec le calendrier natif *iOS / Android*).

Vous pourrez adapter et compléter les fonctionnalités ci-dessus selon votre inspiration.

Les fonctionnalités avancées apportent des points bonus à l'évaluation.

Exigences fonctionnelles et techniques

- **Structuration du projet en plusieurs fichiers** (1 widget = 1 fichier) et **dossiers** (organisés par *feature* ou par aspect technique),
- **Emploi de classes *Dart* pour la gestion des services métier**,
- **Navigation entre les différents écrans de l'application avec un système de routes** (*Sign Up, Sign In, Master / Details*),
- **Gestion d'état applicatif local** (*Stateless / Stateful*),
- **Gestion d'état applicatif global avec un store de données** (*Provider*),
- **Stockage des variables d'environnement dans un fichier ".env"**,

- **Absence de messages d'erreur ou d'avertissement** dans l'onglet "*problèmes*" de votre IDE, respect des conventions de nommage fixées par *Dart* et *Flutter*...
- **Développement de l'application en priorité à destination d'un terminal mobile de type smartphone, orienté en mode portrait, sur la plateforme *iOS* et/ou *Android*** (plateformes web et desktop non prioritaires).

Exigences ergonomiques

Vous apporterez un **soin particulier aux aspects UI / UX** de l'application afin de proposer la meilleure expérience possible à l'utilisateur :

- simplicité,
- intuitivité,
- rapidité,
- respect des standards ergonomiques mobiles / tactiles.

Packages recommandés

(Liste non-exhaustive)

Pour ce faire, **vous vous inspirerez des applications faisant référence dans le domaine** (ex: *Rappels*, *Microsoft To do*, *Todoist*, *Evernote*, *Any.do*, ...).

- **Navigation** : https://pub.dev/packages/go_router
- **Store de données** : <https://pub.dev/packages/provider>
- **Client HTTP** : <https://pub.dev/packages/dio>
- **Variables d'environnement** : https://pub.dev/packages/flutter_dotenv
- **Personnalisation du *Splash Screen*** : https://pub.dev/packages/flutter_native_splash
- **Personnalisation de l'*App Icon*** : https://pub.dev/packages/flutter_launcher_icons
- **Préférences de l'application** : https://pub.dev/packages/shared_preferences
- **Chiffrement des données stockées dans les préférences de l'application** : https://pub.dev/packages/encrypt_shared_preferences
- **Base de données locale *SQLite*** : <https://pub.dev/packages/sqlite>
- **Base de données locale *NoSQL*** : <https://pub.dev/packages/hive>
- **Détection de la connexion au réseau** : https://pub.dev/packages/connectivity_plus
- **Caméra** : <https://pub.dev/packages/camera>
- **Carte géographique interactive** : https://pub.dev/packages/flutter_map

Modalités de réalisation

- **Travail individuel.**
- **Code personnel** (usage modéré de l'IA générative).
- Rendu sous forme de dépôt Git partagé à alex@shrp.dev / shrp777 :
 - Renseigner l'identité de l'auteur de l'application dans un fichier README.md à la racine du projet.
 - Associer 1 branche Git pour chaque étape de TD.
 - Décrire le code réalisé à chaque étape de TD dans le fichier README.md
- Délai de rendu : **mercredi 19/6/2024, 23h59 max.**

Critères d'évaluation

- Etat fonctionnel de l'application,
- Respect des consignes,
- Qualité du code,
- Qualité *UI / UX*,
- Participation, assiduité et comportement respectueux pendant les séances de cours.