

Guide pratique pour la rédaction d'un rapport de Projet de Fin d'Études (PFE)

Pour les étudiants des filières d'Ingénierie Digitale et Intelligence Artificielle à l'école EIDIA

Prof. MERYAM ELMOUHTADI
Université Euromed de Fès

14 mai 2025

Table des matières

1	Remerciements	3
2	Résumé	3
3	Abstract	3
4	Introduction	3
4.1	Objectifs du rapport	3
4.2	Structure du rapport	3
5	Présentation de l'organisme d'accueil	4
5.1	Fiche d'identité	4
5.2	Historique et missions	4
5.3	Structure organisationnelle	4
6	État de l'art	4
6.1	Concepts théoriques	4
6.2	Technologies comparées	4
6.3	Choix technologiques	6
7	Analyse des besoins	6
7.1	Expression du besoin	6
7.2	Spécifications fonctionnelles	6

7.3	Spécifications techniques	6
7.4	Diagrammes de cas d'utilisation (facultatif)	6
8	Conception	6
8.1	Architecture du système	6
8.2	Modèles UML	7
8.3	Modèle de données (En fonction du projet)	7
8.4	Choix techniques	7
9	Réalisation	7
9.1	Environnement de développement	7
9.2	Développement des modules	7
9.3	Interfaces utilisateur	7
9.4	Problèmes rencontrés et solutions apportées	7
10	Tests et validation	7
10.1	Types de tests réalisés	8
10.2	Résultats des tests	8
10.3	Éventuels dysfonctionnements	8
11	Conclusion et perspectives	8
11.1	Bilan du projet	8
11.2	Compétences développées	8
11.3	Limites du projet	8
11.4	Perspectives d'amélioration	8
12	Annexes	9
13	Bibliographie & Webographie	9

Table des figures

Liste des tableaux

1	Comparatif des outils de développement d'applications mobiles	5
---	---	---

1. Remerciements

Cette section permet d'exprimer votre gratitude à l'égard des personnes qui vous ont accompagné dans la réalisation du projet. Il est d'usage de remercier l'organisme d'accueil, l'encadrant professionnel, l'encadrant universitaire ainsi que toute autre personne ayant contribué de manière significative à votre réussite.

2. Résumé

Le résumé présente de manière concise le sujet du PFE, ses objectifs, la méthodologie utilisée, les résultats obtenus et les conclusions principales. Il doit permettre au lecteur de comprendre l'essentiel du projet en quelques lignes.

3. Abstract

Cette section est la traduction anglaise fidèle du résumé. Elle est nécessaire pour la lisibilité internationale du rapport.

4. Introduction

Ce guide est destiné aux étudiants de la dernière année des filières d'Ingénierie Digitale et Intelligence Artificielle réalisant leur projet de fin d'études (PFE). Il propose une structure claire, des conseils rédactionnels et des exemples concrets pour rédiger un rapport académique rigoureux, conforme aux attentes pédagogiques.

4.1. Objectifs du rapport

Détaillez les objectifs fixés au début du projet, qu'ils soient techniques, organisationnels ou méthodologiques. Les points suivants sont à prendre en considération :

- ✧ Documenter le projet réalisé en expliquant sa démarche, ses résultats et ses limites.
- ✧ Valoriser les compétences acquises (techniques, rédactionnelles, organisationnelles).
- ✧ Démontrer une capacité de structuration, d'analyse et de synthèse.

4.2. Structure du rapport

Pour garantir la clarté, la cohérence et la qualité académique du document, il est recommandé de suivre une structure générale composée des sections suivantes, chacune ayant un rôle précis dans la présentation du projet de fin d'études :

1. Page de garde

2. Remerciements
3. Résumé / Abstract
4. Table des matières
5. Introduction
6. Présentation de l'organisme d'accueil
7. État de l'art
8. Analyse des besoins
9. Conception
10. Réalisation
11. Tests et validation
12. Conclusion & perspectives
13. Annexes
14. Bibliographie & Webographie

5. Présentation de l'organisme d'accueil

5.1. Fiche d'identité

Décrivez l'entreprise ou le laboratoire : nom, secteur, statut juridique, effectif, localisation.

5.2. Historique et missions

Présentez brièvement l'historique, les valeurs et les activités principales.

5.3. Structure organisationnelle

Ajoutez un organigramme ou une description du service d'accueil.

6. État de l'art

6.1. Concepts théoriques

Présentez les notions nécessaires pour comprendre le projet (ex. : Machine learning, POO, architectures web, API REST, mobile, ...).

6.2. Technologies comparées

Comparez les outils, frameworks, plateformes possibles.

Outil	Type de développement	Langages principaux	Avantages clés
Android Studio	Natif (Android)	Java, Kotlin	IDE officiel Android, outils de débogage avancés, bonne documentation
Xcode	Natif (iOS)	Swift, Objective-C	Intégration avec iOS, simulateurs puissants, interface conviviale
Flutter	Cross-platform	Dart	Interface fluide, hot reload, un seul code pour Android/iOS
React Native	Cross-platform	JavaScript, React	Utilise des composants natifs, grande communauté, code JavaScript réutilisable
Ionic	Hybrid (WebView)	HTML, CSS, JavaScript	Facile à apprendre, basé sur Angular, rapide à prototyper
Xamarin	Cross-platform	C#	Intégré à l'écosystème Microsoft, accès aux API natives, code partagé élevé

TABLE 1 – Comparatif des outils de développement d'applications mobiles

6.3. Choix technologiques

Justifiez les choix retenus en fonction des critères du projet. (Exemple pour application web et mobile)

7. Analyse des besoins

L'analyse des besoins est une étape clé qui vise à définir clairement ce que le système doit accomplir, tant sur le plan fonctionnel que technique. Elle permet de cadrer le périmètre du projet en identifiant les attentes des utilisateurs et les contraintes du système. Les besoins seront présentés ci-dessous en distinguant les aspects fonctionnels des exigences non fonctionnelles.

7.1. Expression du besoin

Présentez la problématique concrète à résoudre.

7.2. Spécifications fonctionnelles

Listez les fonctionnalités prévues, sous forme de tableau ou user stories.

7.3. Spécifications techniques

Décrivez les contraintes de sécurité, compatibilité, performance, etc.

7.4. Diagrammes de cas d'utilisation (facultatif)

Utilisez UML pour illustrer les interactions principales.

8. Conception

La phase de conception consiste à traduire les besoins identifiés en une architecture logique et technique du système. Elle permet de définir la structure de l'application, les composants clés, ainsi que les interactions entre les différentes entités. Cette section présente les éléments de conception sous forme d'items et de schémas (diagrammes UML, architecture, etc.), afin de poser les bases solides du développement à venir.

8.1. Architecture du système

Expliquez l'architecture retenue (MVC, microservices...).

8.2. Modèles UML

Incluez des diagrammes de classes, de séquence, d'activités.

8.3. Modèle de données (En fonction du projet)

Présentez le MCD et éventuellement le MLD.

8.4. Choix techniques

Détaillez les langages, frameworks, bibliothèques retenus.

9. Réalisation

La phase de réalisation correspond à la mise en œuvre concrète du projet, conformément aux spécifications définies lors de la conception. Elle comprend le développement des différentes fonctionnalités, l'intégration des composants, ainsi que la mise en place des tests. Cette section détaille les principales étapes de l'implémentation, les outils utilisés, les difficultés rencontrées et les solutions apportées.

9.1. Environnement de développement

Indiquez les outils utilisés : IDE, gestion de versions, serveurs...

9.2. Développement des modules

Expliquez les étapes de codage et montrez des extraits significatifs.

9.3. Interfaces utilisateur

Insérez des captures d'écran annotées des pages ou interfaces réalisées.

9.4. Problèmes rencontrés et solutions apportées

Soyez transparent sur les obstacles et les résolutions.

10. Tests et validation

La phase de tests et validation vise à s'assurer que l'application développée fonctionne conformément aux spécifications initiales et qu'elle répond aux attentes exprimées lors de l'analyse des besoins. Elle permet d'identifier d'éventuels dysfonctionnements, de vérifier la robustesse du système et de garantir sa qualité avant livraison. Cette section présente les

types de tests réalisés, leur méthodologie, les résultats obtenus et les éventuelles corrections apportées.

10.1. Types de tests réalisés

Tests unitaires, fonctionnels, d'intégration...

10.2. Résultats des tests

Présentez les résultats sous forme de tableau (cas de test, résultat attendu, résultat obtenu).

10.3. Éventuels dysfonctionnements

Indiquez les problèmes rencontrés lors des tests, leur contexte d'apparition, leur impact sur le projet ainsi que les solutions apportées ou envisagées.

11. Conclusion et perspectives

La conclusion synthétise l'ensemble du travail réalisé et met en lumière les apports personnels et professionnels du projet.

11.1. Bilan du projet

Rappelez brièvement les objectifs et ce qui a été accompli.

11.2. Compétences développées

Listez les acquis techniques, humains, organisationnels.

11.3. Limites du projet

Décrivez honnêtement les limites rencontrées.

11.4. Perspectives d'amélioration

Proposez des axes d'évolution ou d'optimisation.

12. Annexes

Les annexes regroupent l'ensemble des éléments complémentaires utiles à la compréhension du projet, tels que des extraits de code, des diagrammes, ou des documents techniques.

- ✧ Code source commenté (parties importantes uniquement).
- ✧ Schémas supplémentaires.
- ✧ Manuel d'installation ou d'utilisation.

13. Bibliographie & Webographie

La bibliographie recense l'ensemble des ouvrages, articles scientifiques et documents académiques consultés pour la réalisation du projet, en respectant une norme de citation cohérente (ex. APA, IEEE). Exemples :

1. A. Turing, *Computing Machinery and Intelligence*, Mind, 1950.

La webographie regroupe les sources en ligne utilisées, telles que sites web, tutoriels ou documentations officielles, en précisant systématiquement la date de consultation pour garantir la traçabilité des informations.

1. Documentation officielle de Django : <https://docs.djangoproject.com> ; visité le : 14 mai 2025.