

# Synthèse des Travaux Réalisés

## Introduction

Dans le cadre de ce projet, nous avons conçu une application permettant la gestion du temps passé sur l'ordinateur. Le système offre des fonctionnalités comme le suivi des applications utilisées, la gestion des notifications, et la définition d'objectifs de productivité. Cette synthèse présente les étapes réalisées, les choix de conception effectués, ainsi que les livrables produits.

## 1. Exigences Fonctionnelles

- Suivi du temps d'utilisation des applications.
- Blocage des applications définies par l'utilisateur.
- Notifications envoyées lorsque les limites de temps sont atteintes.
- Définition d'objectifs de productivité et gestion des tâches associées.
- Génération de rapports d'utilisation (quotidien, hebdomadaire, mensuel).

## 2. Démarche Suivie

- **Analyse des besoins** : identifiez les fonctions clés en fonction des exigences de l'application.
- **Conception UML** :
  - Création des **diagrammes de classes** pour modéliser les entités et leurs relations.
  - Élaboration d'un **diagramme de cas d'utilisation** pour décrire les interactions utilisateur-système.
- **Prototypage** : Réalisation des maquettes pour visualiser l'interface utilisateur.
- **Développement** : Mise en place de la structure initiale du code en suivant une architecture orientée objet.
- **Documentation** : Élaboration des documents détaillant les exigences, la démarche, et les choix technologiques.

### 3. Représentation Graphique

- **Diagramme de classes :**
  - Modèle complet des entités du système, incluant des relations fortes telles que la composition entre les objectifs et leurs tâches.
  - Les diagrammes ont été créés avec **Lucidchart** pour une représentation claire et professionnelle.
- **Diagramme de cas d'utilisation :**
  - Illustration des interactions utilisateur-système, telles que l'ajout d'une application, la configuration des notifications, et la gestion des objectifs.
  - Le diagramme met en évidence les dépendances entre les cas d'utilisation (e.g., inclusion entre "Définir une limite d'utilisation" et "Ajouter une application").

### 4. Prototypes Visuels

- Les maquettes fournies (voir document joint) montrent :
  - L'interface principale pour le suivi du temps.
  - Les écrans de configuration des limites et des notifications.
  - La gestion des objectifs de productivité et des rapports.

### 5. Outils Utilisés

- **Lucidchart** : Création des diagrammes UML, incluant les diagrammes de classes et le diagramme de cas d'utilisation.
- **Figma** : Création des maquettes pour les interfaces utilisateur.
- **GitHub** : Gestion du code source et des documents partagés.

## 6. Structure Initiale du Code

- Une structure modulaire respectant les principes de l'orienté objet :
  - **Classes principales** : GestionnaireApplication, SuiviTemps, PréférencesUtilisateur, etc.
  - **Gestion des relations** : Utilisation correcte des compositions et des associations.
- Les fichiers incluent des commentaires explicatifs pour chaque classe et méthode.

## 7. Choix de Conception

- **Approche orientée objet** pour modularité et extensibilité.
- **Relation de composition** entre ObjectifProductivite et Tache, garantissant une gestion stricte des dépendances.
- **Notifications configurables** pour permettre à l'utilisateur de contrôler les alertes.

## 8. Résultats

- Diagrammes UML réalisés avec **Lucidchart**, incluant le **diagramme de classes** et le **diagramme de cas d'utilisation**.
- Prototypes visuels conçus avec Figma pour refléter les interactions clés.
- Code initial structuré avec les relations correctement définies.
- Documentation complète des choix effectués.

## Conclusion

Cette première itération pose une base solide pour le développement de l'application. Les livrables produits, incluant diagrammes UML, maquettes et structure de code, répondent aux attentes du projet. Les prochaines étapes consisteront à enrichir les fonctionnalités et à intégrer le retour des utilisateurs sur les prototypes.