

## Vision synthétique

Le Solopreneur OS se veut un **tableau de bord personnel souverain**, orienté vers l'action et l'autonomie. Son objectif principal est d'aider le créateur d'entreprise à « récupérer les opportunités perdues » dans sa gestion quotidienne, en transformant chaque tâche réalisée en valeur concrète <sup>1</sup>. Il n'est pas vendu comme un simple outil technique, mais comme un **système garantissant un résultat mesurable** (par exemple, « ne manquer aucune tâche importante »), à l'instar de l'approche de Growify qui se présente comme une « machine anti-perte » <sup>1</sup>. Sa mission est donc de centraliser la capture d'idées (inbox), la planification (projets, tâches, calendrier), l'exécution et le suivi (progression journalière, KPI business) en un même espace local. La promesse du système est d'augmenter la productivité et la souveraineté de l'utilisateur en lui fournissant un workflow cohérent et gamifié, aligné avec ses valeurs d'autonomie et d'efficacité (profil « idéaliste empathique » INFJ) <sup>1</sup>.

## Architecture modulaire

L'OS adopte une **architecture modulaire** claire, chaque page étant un module autonome interconnecté. Les routes existantes de l'application indiquent déjà les modules clés : `/today` (vue du jour), `/inbox`, `/tasks` (Kanban), `/projects`, `/notes`, `/calendar`, `/business` <sup>2</sup>. Par exemple, la page **Aujourd'hui (Today)** affiche les tâches prioritaires du jour avec une barre de progression ( **Progression du jour**), tandis que le **Tableau des tâches (Tasks Board)** offre une vue Kanban multi-colonnes (À faire, En cours, Bloqué, Terminé) pour organiser l'ensemble du backlog. Les projets (PARA) servent de conteneur pour regrouper les tâches, et un module Notes permet de documenter idées et retours. Chaque composant est lié : une tâche peut provenir de l'inbox, être rattachée à un projet, avec une échéance sur le calendrier, et contribuer aux indicateurs business. Cette conception s'inspire d'une approche « **offre modulaire** » où la valeur perçue est structurée en blocs indépendants <sup>3</sup>.

## Logique de persistance quotidienne

Le système enregistre l'avancement au jour le jour. Les tâches porteuses d'un flag "Today" ou dont la date d'échéance est aujourd'hui remontent automatiquement dans la vue **Aujourd'hui**. Chaque tâche terminée déclenche un enregistrement de complétion (on peut imaginer une table `task_completions`) contenant la date d'achèvement, la durée réelle et d'autres métadonnées. Sur chaque jour, on calcule le pourcentage de tâches accomplies (ex. « 5 sur 8 terminées → 62% ») pour alimenter la **progression journalière** affichée <sup>4</sup>. En cas d'erreur (tâche oubliée ou marquée par mégarder), l'utilisateur peut corriger rétroactivement la complétion du jour, et le système mettra à jour les statistiques (tâches complètes, %). Ainsi, la logique de persistance combine un journal quotidien de complétion avec la table des tâches, garantissant un suivi précis des réalisations sans perdre les données passées. Par défaut, les tâches non terminées sont reportées automatiquement (ou replanifiées) le jour suivant, évitant les omissions définitives.

# Logique de gamification

Pour motiver l'utilisateur, l'OS intègre un système de **points d'expérience (XP)** et de niveaux. Chaque tâche complétée rapporte des XP proportionnels à sa priorité ou durée estimée : par exemple, une tâche courte apporte 10 XP, une tâche longue 20 XP. On peut définir un niveau par palier d'XP (ex. 100 XP = niveau 2), affichant un avatar ou une jauge de progression. Des **défis journaliers** invitent à viser un quota d'XP ou un objectif de tâches (ex. « 50 XP par jour »). Des **récompenses symboliques** (badges, titres, rituel d'évaluation) ponctuent les jalons importants. Par exemple, terminer 5 jours d'affilée au-dessus de 80% de progression active un badge « Régularité ». La logique encourage aussi des **rituels de planification** : un pop-up de début de journée fixe les priorités (Kick-off du jour), et une synthèse de fin de journée consigne les leçons (morning/ evening routine), intégrant des rappels de pause ou déconnexion pour renforcer la discipline personnelle.

# Interfaçage et UX

L'interface doit refléter l'esprit de souveraineté : clair, épuré et adaptable. On utilisera un design "**local-first**" : l'application fonctionne en local (PWA ou desktop), le thème sombreclair synchronisé avec le système, et les données sont stockées sur l'appareil (sans obligation de cloud). Les écrans sont prioritaires responsive pour mobile, tablette et desktop. L'**Inbox** est accessible en un clic pour saisir rapidement une idée (champ minimaliste). La page **Aujourd'hui** affiche clairement la progression avec une barre colorée et un décompte « X sur Y tâches »<sup>4</sup>. Les éléments terminés sont visuellement atténus, avec option « masquer/afficher complétées »<sup>5</sup>. Le **Kanban** offre du drag-and-drop intuitif avec une limite WIP (max 3 tâches en cours) alertée visuellement comme aujourd'hui dans le code<sup>6</sup>. Les filtres (par projet, tags) simplifient la vue si nécessaire. Les icônes et couleurs sont cohérentes pour chaque module (⌚ aujourd'hui, ⌚ tâches, projets, calendrier, ↗ business, etc.), renforçant l'organisation visuelle. L'UX priviliege la **cohérence** (mêmes boutons d'action, mêmes notifications toast pour succès/erreur) et la **personnalisation** (possibilité de définir des raccourcis, thème).

# Vision long terme

Le Solopreneur OS est construit pour évoluer : il sera **extensible** avec des plugins ou intégrations (e.g. synchronisation facultative avec Google Calendar, import/export CSV, connexion à un assistant vocal). L'architecture doit être ouverte aux IA : par exemple un assistant intégré qui suggère la priorisation automatique des tâches (« quelle tâche est la plus rentable aujourd'hui ? ») ou qui génère des résumés de performance. Le système encourage l'**autonomie** maximale : possibilité de fonctionner offline complet pour une autonomie totale. Il intègre des **sauvegardes locales** et exportations pour déconnexion ponctuelle. Sur le long terme, on peut envisager une déconnexion intelligente : après avoir accompli les tâches de la journée et atteint le quota d'XP, l'application peut suggérer un passage en mode « privé » ou proposer des rituels de clôture (journal, méditation guidée). En somme, le Solopreneur OS se conçoit comme une **base durable** : chaque module (Inbox, Today, Kanban, etc.) peut être amélioré ou remplacé indépendamment, l'ensemble restant cohérent. Cette vision ambitieuse repose sur le principe du local-first et sur la gamification pour cultiver l'engagement de l'utilisateur dans son « projet de vie » personnel, tout en gardant la flexibilité d'adapter l'outil aux besoins futurs.

**Sources :** D'après les documents internes (stratégie Growify, spécifications techniques) et l'examen du code existant, notamment les routes définies<sup>2</sup> et les exigences de la vue *Aujourd'hui* (tâches du jour et progression)<sup>4</sup>, ainsi que la promesse produit claire exprimée dans la stratégie Growify<sup>1</sup> et

l'approche modulaire de son offre <sup>3</sup>. Ces éléments ont guidé la conception modulaire et fonctionnelle proposée ci-dessus.

---

<sup>1</sup> <sup>3</sup> **Synthèse Stratégique \_ Le Projet Growify.pdf**

file://file-G9GLoaz73DmGkdzAEt73wm

<sup>2</sup> **web.php**

<https://github.com/Yacoubbeltanivelle/TableauDeBord/blob/d5751a55cd4d4f49fc83a914792621f6befc1e31/app/routes/web.php>

<sup>4</sup> **SPEC.md**

<https://github.com/Yacoubbeltanivelle/TableauDeBord/blob/d5751a55cd4d4f49fc83a914792621f6befc1e31/docs/SPEC.md>

<sup>5</sup> **Today.tsx**

<https://github.com/Yacoubbeltanivelle/TableauDeBord/blob/d5751a55cd4d4f49fc83a914792621f6befc1e31/app/resources/js/Pages/Today.tsx>

<sup>6</sup> **TasksBoard.tsx**

<https://github.com/Yacoubbeltanivelle/TableauDeBord/blob/d5751a55cd4d4f49fc83a914792621f6befc1e31/app/resources/js/Pages/TasksBoard.tsx>