**🎯 Contexte général du projet**

**Tu travailles sur un système de gestion budgétaire à l’UCAD / ESP, avec comme objectif principal :**

**👉 Générer automatiquement un projet de budget consolidé (recettes et dépenses) à partir des besoins exprimés dans chaque département, en respectant une nomenclature budgétaire standardisée.**

**🧱 Objectifs clés du projet**

* **Faciliter l'expression des besoins dans chaque département.**
* **Permettre aux chefs de département de consolider les besoins pour leur unité.**
* **Offrir au directeur la possibilité de modifier ou valider le projet final (ex. en conférence budgétaire).**
* **Intégrer les données du budget de l’année précédente (proposé et réalisé) pour permettre des analyses de variation.**
* **Respecter la nomenclature budgétaire (70 pour ventes, 718 pour subventions, 60 pour achats, 66 pour salaires, etc.).**
* **Gérer les formations payantes, prestations d’expertise, ou d’autres activités génératrices de revenus.**
* **Fournir une vue d’ensemble claire (tableau de bord) : demandé vs réalisé vs manquant.**

**🔧 Phase 1 – Inception (Faisabilité et cadrage)**

**✅ Objectifs de la phase :**

* **Définir le périmètre du système à développer.**
* **Identifier les acteurs, leurs rôles, et les attentes principales.**
* **Identifier les risques potentiels.**
* **Déterminer la viabilité technique et organisationnelle du projet.**
* **Ébaucher une étude de rentabilité : gains pour l’ESP/UCAD.**

**👥 Acteurs identifiés :**

| **Acteur** | **Rôle principal** |
| --- | --- |
| **Chef de département** | **Saisie et consolidation des besoins** |
| **Directeur** | **Validation finale du budget** |
| **Services techniques** | **Saisie des recettes/dépenses propres** |
| **Responsable budgetaire** | **Génération automatique du projet** |
| **Développeur** | **Réalisation du système** |
| **Comptable / Auditeur** | **Analyse des écarts** |

**📌 Fonctionnalités attendues (exprimées sous forme de cas d’utilisation) :**

* **📝 Exprimer les besoins d’unité pédagogique ou technique.**
* **📊 Visualiser les budgets précédents (proposé, réalisé).**
* **➕ Ajouter ou modifier des lignes budgétaires selon nomenclature.**
* **📑 Générer automatiquement un fichier de projet budgétaire.**
* **📈 Comparer les variations (écart budgétaire, taux de réalisation).**
* **🔄 Réviser le projet à plusieurs niveaux : service, département, direction.**

**⚠️ Risques identifiés :**

* **❌ Non-uniformité des expressions de besoins entre services.**
* **❌ Erreurs de saisie ou confusion sur les codes budgétaires.**
* **❌ Difficulté à accéder aux données de l’année précédente.**
* **❌ Manque de formation des acteurs à l’outil.**

**✅ LIVRABLES DE LA PHASE 1 – INCEPTION (Projet : Gestion budgétaire UCAD/ESP)**

**1. 📃 Cahier des charges fonctionnel simplifié (V1) ✅**

Objectif général :

Mettre en place une application permettant la saisie, la consolidation et la génération automatique du projet de budget annuel de chaque entité (département, direction, service), en respectant la nomenclature UCAD/ESP.

Fonctionnalités fonctionnelles :

- Saisie des besoins par service/département selon nomenclature (recettes, dépenses).

- Accès aux données des années précédentes (budget proposé et réalisé).

- Calcul automatique des écarts (valeur absolue et relative).

- Génération du projet budgétaire (fichier consolidé).

- Gestion de rôles : utilisateur (saisie), chef de département (validation), direction (édition finale).

- Export Excel ou PDF du projet.

Contraintes :

- Respect strict de la nomenclature UCAD (ex : 70 = Ventes, 60 = Achats...).

- Interface simple utilisable par des non-informaticiens.

- Multisession, multi-utilisateur sécurisé.

**2. 🧑‍🤝‍🧑 Liste des acteurs et rôles ✅**

| Acteur | Rôle dans le système |
| --- | --- |
| Chef de service | Saisie des besoins et recettes |
| Chef de département | Consolidation des données du service |
| Directeur | Validation/modification finale du budget |
| Comptable | Analyse des écarts (proposé vs réalisé) |
| Développeur | Conception, implémentation technique |
| Responsable budgetaire | Supervision globale, génération du fichier final |

**3. 📊 Liste des cas d’utilisation priorisés ✅**

| ID | Cas d'utilisation | Priorité |
| --- | --- | --- |
| CU1 | Saisir les besoins d’un département | Haute |
| CU2 | Saisir les recettes prévues | Haute |
| CU3 | Accéder aux budgets passés | Haute |
| CU4 | Calculer et afficher les écarts | Moyenne |
| CU5 | Valider ou refuser un budget consolidé | Haute |
| CU6 | Générer le projet de budget au format Excel/PDF | Haute |
| CU7 | Modifier le budget lors de la conférence budgétaire | Moyenne |

**4. 🧠 Liste des rubriques budgétaires (partielle) ✅**

Extrait de la nomenclature du document :

| Code | Intitulé | Type |
| --- | --- | --- |
| 70011 | Droits d'inscription administrative | Recette |
| 7065 | Cessions de services | Recette |
| 60411 | Consommables informatiques | Dépense |
| 60512 | Eau minérale | Dépense |
| 66111 | Salaires PATS | Dépense |
| 2442 | Matériel informatique (Investissement) | Dépense |

**5. 🏛️ Esquisse de l’architecture du système ✅**

**Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.**

**6. 💰 Ébauche de rentabilité (impact) ✅**

| **Aspect** | **Gains attendus** |
| --- | --- |
| Temps | Moins de saisie manuelle, travail consolidé |
| Fiabilité | Réduction des erreurs de calcul ou doublons |
| Transparence | Traçabilité des modifications |
| Clarté | Vue consolidée pour la conférence budgétaire |

**🧱 PHASE 2 – ÉLABORATION**

Objectif : définir **l’architecture du système**, affiner les besoins, préparer le terrain pour le développement technique.

**✅ 1. Objectifs de la phase**

* Consolider les besoins fonctionnels (au moins 80 % des cas d’utilisation)
* Définir l’**architecture logicielle**
* Préciser les rôles, interfaces, priorités
* Planifier la **phase de construction**
* Valider la **faisabilité technique**
* Identifier les composants critiques

**📌 2. Besoins fonctionnels détaillés**

| **Code** | **Fonctionnalité** | **Description** |
| --- | --- | --- |
| F01 | Saisie des besoins | L’utilisateur saisit les besoins budgétaires (recettes/dépenses) selon la nomenclature |
| F02 | Accès au budget passé | Permet de consulter les budgets précédents (proposé et réalisé) |
| F03 | Calcul des écarts | Calcul automatique des variations en valeur et en % |
| F04 | Consolidation départementale | Le chef de département fusionne les besoins de ses services |
| F05 | Modification direction | Le directeur peut corriger ou ajuster les lignes budgétaires |
| F06 | Génération automatique du projet de budget | Exporte le fichier final (Excel ou PDF) |
| F07 | Tableau de bord des écarts | Affiche clairement les manques/écarts par service ou rubrique |

**👥 3. Rôles système + flux**

| **Rôle** | **Droits** |
| --- | --- |
| **Utilisateur simple** | Saisie des besoins |
| **Chef de département** | Validation / Consolidation |
| **Direction** | Révision / Édition finale |
| **Comptable / Audit** | Lecture / Analyse |
| **Développeur** | Accès technique global |

**⚙️ 4. Architecture technique proposée**

**🧱 Architecture 3 couches**

1. **Front-end** (interface utilisateur)
   * Technologies : HTML, CSS, JavaScript, Bootstrap
   * Alternatives plus puissantes : Vue.js ou React
   * Interface responsive simple
2. **Back-end** (logique métier & API)
   * Framework : Python Flask ou Django
   * Fonctions : traitement des saisies, calcul des écarts, génération de fichier, autorisations
3. **Base de données**
   * PostgreSQL ou SQLite
   * Tables principales :
     + users
     + departments
     + budget\_lines
     + budget\_history
     + roles
     + rubriques\_nomenclature
4. **Fichiers générés**
   * Export Excel ou PDF
   * Génération avec openpyxl ou pandas côté back-end

**🗂️ 5. Modèle de données simplifié**

| **Table budget\_lines** |
| --- |
| id |
| department\_id |
| rubrique\_code |
| rubrique\_label |
| type (recette/dépense) |
| montant\_propose |
| montant\_realise |
| année |
| user\_id |

| **Table users** |
| --- |
| id |
| nom |
| email |
| mot\_de\_passe |
| role\_id |

| **Table roles** |
| --- |
| id |
| nom (utilisateur, chef, directeur, etc.) |

| **Table rubriques\_nomenclature** |
| --- |
| code |
| libellé |
| type (recette/dépense) |

**📅 6. Planification indicative de la phase suivante (Construction)**

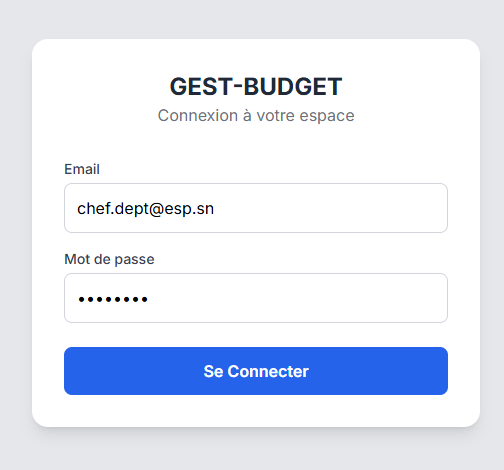
| **Étape** | **Durée estimée** | **Livrable attendu** |
| --- | --- | --- |
| Création de la base de données | 2 jours | Modèle de données fonctionnel |
| Développement de l’interface de saisie | 3-5 jours | Formulaire connecté |
| Gestion des profils utilisateurs | 2 jours | Authentification, rôles |
| Calcul automatique des écarts | 1 jour | Vue des résultats |
| Génération du fichier final | 2 jours | Fichier Excel / PDF |
| Tableau de bord direction | 2 jours | Vue globale, écarts, statistiques |

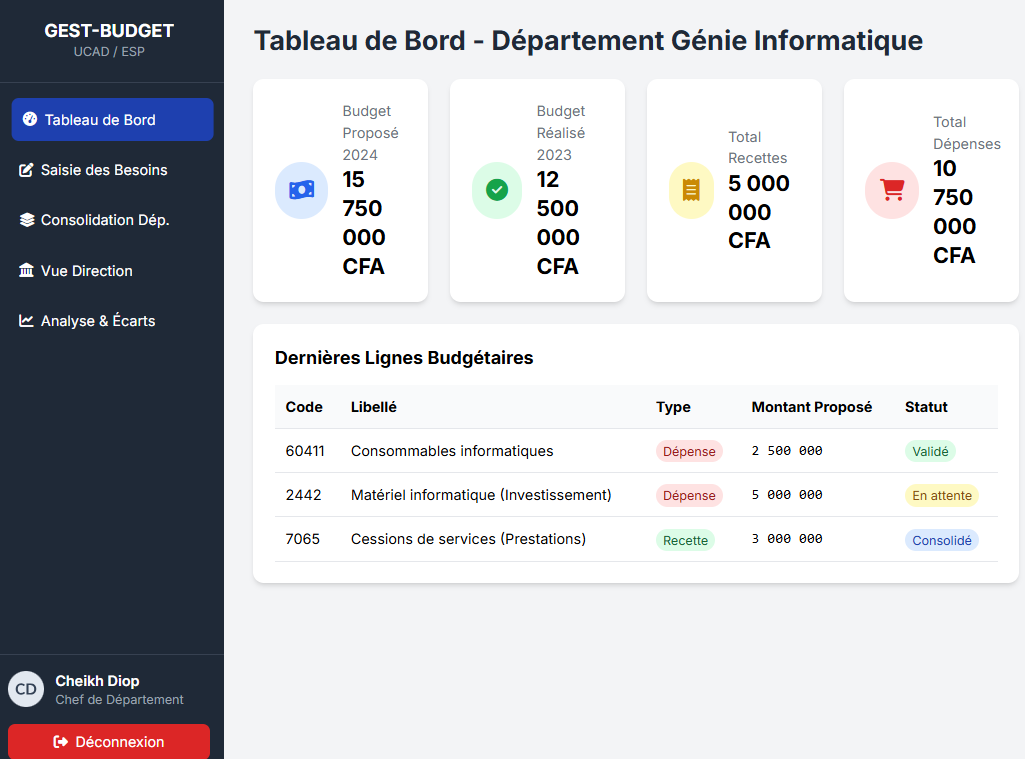
**📤 7. Livrables à produire dans cette phase**

* Diagramme de classes UML 📘

… (à faire)

* Maquette des interfaces 🎨





Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, logiciel, Page web

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, logiciel, Page web

Le contenu généré par l’IA peut être incorrect.

* Modèle de données (format SQL)

Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

* Spécifications techniques détaillées

| **Composant** | **Technologie possible** |
| --- | --- |
| Frontend | HTML/CSS/Bootstrap + JS (ou React) |
| Backend | Python (Flask ou Django) |
| Base de données | PostgreSQL ou SQLite |
| Export | openpyxl, reportlab (PDF) |
| Authentification | JWT ou sessions classiques |

* Cas d’utilisation UML (avec acteurs et flux)

… (à dessiner)