

PLAN PROJET

Document

Version 1.0

01 Février 2022

AWS reStart DLACM1 - Concours 1

Rédigé par **Groupe 2**

Table des matières

Table des matières	i
Liste des tableaux	ii
Liste des figures	ii
1 Introduction.....	1
1.1 Suivi des modifications du document	1
1.2 Résumé	1
1.3 Glossaire.....	2
1.4 Références.....	2
1.5 Aperçu du document.....	2
2 Description du projet.....	2
2.1 Buts et objectifs du projet et résultats opérationnels	2
2.2 Jalons	3
2.3 Produits livrables	4
2.4 Planification et organisation du projet	5
2.5 Risques	6
3 Spécification des exigences	7
3.1 Clarification des exigences du client.....	7
3.1.1 Module 1	7
3.1.2 Module 2	8
3.1.3 Module 3	8
3.2 Exigences fonctionnelles.....	9
3.2.1 Environnement du système	9
3.2.2 Les acteurs.....	10
3.2.3 Les cas d'utilisation	10
3.3 Exigences non fonctionnelles	11
3.3.1 Modules 1 et 2.....	11
3.3.2 Module 3	12
Références bibliographiques.....	13

Liste des tableaux

Tableau 1: Buts, objectifs et résultats opérationnels	3
Tableau 2: Jalons du projet	4
Tableau 3: Produits livrables	4
Tableau 4: Matrice des responsabilités (RACI)	6
Tableau 5: Les risques du projet.....	7

Liste des figures

Figure 1 : Diagramme de Gantt du projet	5
Figure 2 : Diagramme des packages.....	10

1 Introduction

1.1 Suivi des modifications du document

L'objectif de la présente section est de contrôler l'élaboration et l'application des modifications apportées à la charte de projet.

Numéro de révision	Date de publication	Auteur(s)	Brève description du changement
0.1	29-01-2022	Aristide	Création du document.
0.2	31-01-2022	Aristide Aubin Patricia Vanessa	Clarification des éléments de la charte du projet et alignement sur les spécifications, l'organisation et la planification du projet
1.0	01-02-2022	Aristide	Finalisation du document

1.2 Résumé

Ce projet s'inscrit dans le cadre de notre formation à ODC qui vise à donner aux apprenants des compétences en administration système et en cloud computing à l'issue du programme AWS ReStart. Pour cela, notre formateur M. Ismael GADJI joue le rôle de client nous confie un projet composé de deux grandes parties. La première partie du projet est constituée des deux premiers modules dans le document que nous a remis le client et a pour objectif de faciliter et d'automatiser la gestion des utilisateurs et de leurs données. La seconde partie, constitué du troisième module consiste à réaliser l'architecture technique d'une infrastructure cloud suivant les exigences spécifiées par le client et à faire un déploiement de cette infrastructure sur AWS.

L'équipe projet devra développer les solutions attendues en respectant le délai fixé du 9 février et en tenant en compte certains risques tels que l'indisponibilité d'un ou plusieurs membres du l'équipe projet.

1.3 Glossaire

Terme	Définition
AWS	Amazon Web Services
Cloud	Plateforme qui offre des services ou des ressources informatiques via internet ou sur un réseau privé.
Cloud computing	
Fichier .csv	Fichier structuré dont les éléments y sont séparés par des points virgules
ODC	Orange Digital Center
RACI	Responsible, Accountable, Consulted, Informed

1.4 Références

<Documents de références utilisés>

1.5 Aperçu du document

La prochaine section, « Description du projet », de ce document donne un aperçu des fonctionnalités du produit. Il décrit les exigences informelles et sert à établir un contexte pour la spécification des exigences techniques dans la troisième section. La troisième section de ce document quant à elle, intitulée « Spécification des exigences », est principalement destiné aux développeurs et décrit en termes techniques les détails des fonctionnalités du produit. Les deux sections du document décrivent le même produit logiciel dans son intégralité, mais sont destinées à des publics différents et utilisent donc un langage différent.

2 Description du projet

2.1 Buts et objectifs du projet et résultats opérationnels

Le tableau suivant décrit les buts du projet en les liant à des objectifs mesurables. En plus, y sont inclus des résultats opérationnels qui découlent des buts et des objectifs connexes.

Tableau 1: Buts, objectifs et résultats opérationnels

N°	Buts	Objectifs	Résultats opérationnels
1	Faciliter la gestion des utilisateurs et de leurs données	<ul style="list-style-type: none"> • Fournir une interface simplifiée pour gérer les utilisateurs (manuellement et dynamiquement) • Automatiser la création simultanée de plusieurs utilisateurs • Automatiser la gestion du cycle de vie des données des utilisateurs supprimés 	<ul style="list-style-type: none"> • Un programme basé sur des menus interactifs qui permet de vérifier un utilisateur, ajouter un ou plusieurs utilisateurs, activer/désactiver et supprimer un compte d'utilisateur. • Archiver automatiquement le dossier personnel d'un utilisateur supprimé et supprimer l'archive 30 jours après la suppression du compte
2	Garantir une haute disponibilité des données afin d'assurer une continuité des activités en cas de sinistre	<ul style="list-style-type: none"> • Redondier périodiquement les dossiers principaux des utilisateurs sur un autre serveur 	<ul style="list-style-type: none"> • Archiver et sauvegarder tous les 7 jours dans un dossier partagé sur un serveur distant chaque dossier personnel des utilisateurs
3	Mettre en place une infrastructure cloud AWS qui répond aux contraintes de criticité des données, de sécurité et de disponibilité en ligne de certains serveurs.	<ul style="list-style-type: none"> • Assurer une forte redondance de l'infrastructure • Créer des sous réseaux avec routages approprié et prenant en compte le nombre d'hôtes 	<ul style="list-style-type: none"> • Proposer une architecture qui illustre tous les éléments de l'infrastructure, les différents sous réseaux, le routage et les différentes redondances dans d'autres régions • Implémenter l'infrastructure sur AWS • Ressortir la table de routage

2.2 Jalons

La section suivante indique les principaux événements dans l'évolution du projet. Ce sont les grandes étapes de notre projet.

Tableau 2: Jalons du projet

Jalon du projet	Description	Date prévue
1. Validation du plan projet	Rédaction du plan projet et validation du document par le client et l'équipe projet	Lundi 31/01/2022
2. Validation du module 1	Livraison des scripts du module 1 et test par les équipes de développement	Mercredi 02/02/2022
3. Validation du module 2	Livraison du script du module 2 et test par les équipes de développement	Jeudi 03/02/2022
4. Validation du module 3	Réalisation de l'architecture et implémentation sur AWS	Vendredi 04/02/2022
5. Livraison de la version Bêta	Réalisation de tous les scénarios de test et livraison d'une première version des deux premiers modules au client	Vendredi 04/02/2022
6. Livraison de la solution et clôture du projet	Corrections des bugs, livraison des solutions finales et de la documentation. Soumission du projet	Lundi 07/02/2022

2.3 Produits livrables

Les principaux produits livrables que le projet doit produire en vue d'atteindre les objectifs fixés sont :

Tableau 3: Produits livrables

Premier produit livrable du projet	Version Bêta
Description :	Livraison d'une première version du module 1 et du module 2 au client
Critères d'acceptation :	Feedback du client sur la version Bêta (Validation de la version / Retour sur les points à corriger)
Échéance :	05 Février 2022
Deuxième produit livrable du projet	Solution finale
Description :	Livraison des solution finales des trois modules au client. Rapport du projet qui décrit en détail les actions menés pour la réalisation du projet. étayé par des captures d'écran qui a la lecture lui permettront de prendre la main sur l'application

Premier produit livrable du projet	Version Bêta
Critères d'acceptation :	Réception et validation des solutions finales et de la documentation par le client.
Échéance :	07 Février 2022

2.4 Planification et organisation du projet

Le diagramme suivant présente les différentes tâches constitutives du projet et leurs échéances respectives.

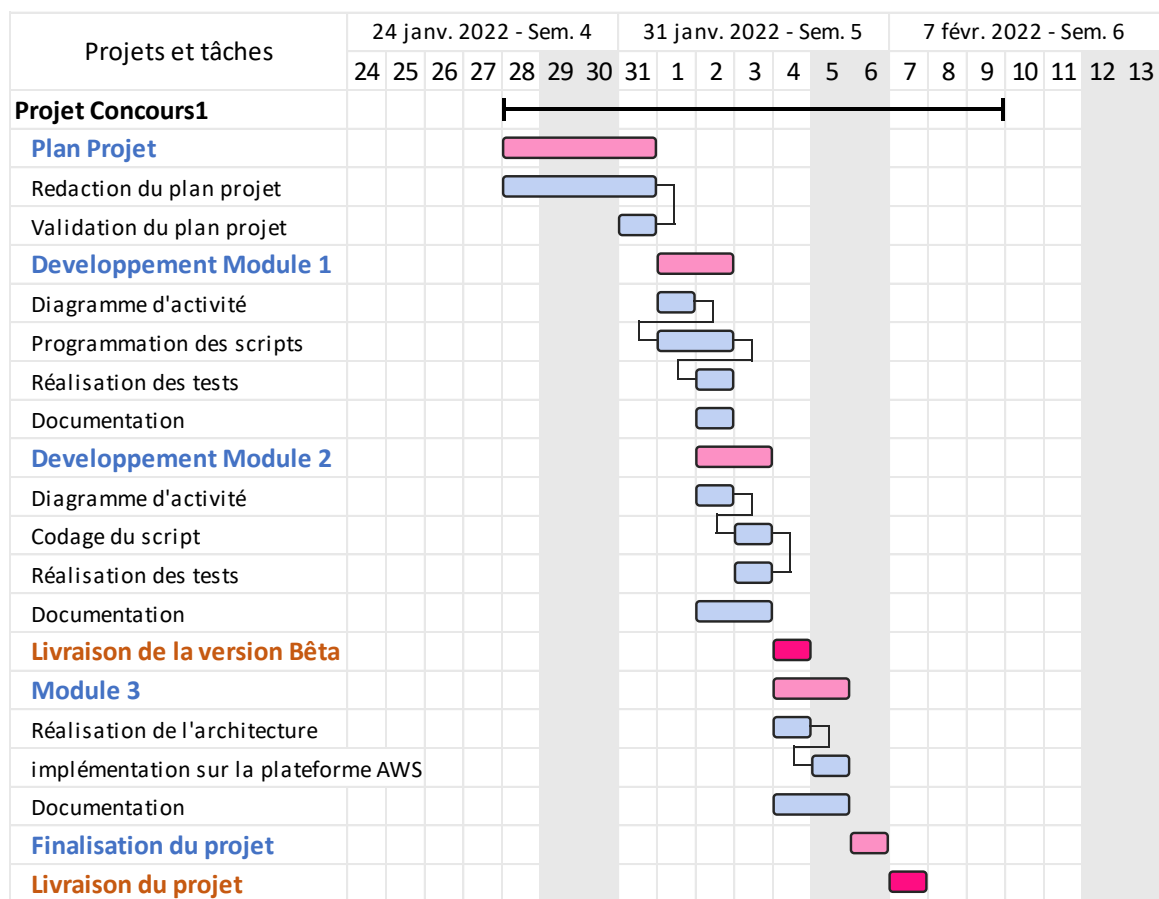


Figure 1 : Diagramme de Gantt du projet

La définition des responsabilités dans le projet a été faite suivant le modèle RACI qui indique les rôles et les responsabilités des intervenants au sein de chaque processus et activité en indiquant les responsables (R : responsable), celui qui rend compte (A :

accountable), ceux qu'on consulte (C : consulted) et ceux qu'on tient informés (I : informed)

Le tableau suivant représente la matrice des responsabilités du projet :

Tableau 4: Matrice des responsabilités (RACI)

		Aristide	Aubin	Patricia	Vanessa	M. Ismael
Plan projet		A/R	C	C	C	C
Module 1	Diagramme d'activité du programme	A	R	C	C	
	Codage du programme	A	R	C	R	
	Réalisation et documentation des tests	C	A/R	C	R	
Module 2	Diagramme d'activité	A	R	C	C	
	Codage du script	A	C	R	R	
	Réalisation et documentation des tests	C	R	R	A	
Module 3	Réflexion et croquis de l'architecture technique	A/R	R	R	R	
	Réalisation de l'architecture technique	A/R	R	R	R	
	Implémentation de l'architecture	A	R	R	R	
Test de la version Bêta		A	I	I	I	R
Documentation		A/R	C	C	C	
Finalisation et livraison		A/R	I	I	I	I

2.5 Risques

Cette partie récence les principaux risques qui ont été identifiés dès le début du projet.

Tableau 5: Les risques du projet

N°	Description du risque	Probabilité (faible, moyenne et élevée)	Incidence (faible, moyenne et élevée)	Atténuation prévue
1.	Indisponibilité d'un ou plusieurs membres de l'équipe projet	Moyenne	Elevé	<ul style="list-style-type: none"> • Au moins deux responsables par tâches, aucune tâche ne doit reposer sur une seule personne. • Mise à jour quotidienne et partage de toutes les informations concernant le projet dans un groupe Whatsapp afin que tout l'équipe soit au même niveau d'information
2.	Retard dans l'implémentation des modules et risque de ne pas livrer toutes les fonctionnalités attendues	Moyenne	Elevé	<ul style="list-style-type: none"> • Deux jours de marge totale ont été prévue dans la planification des livrables • Se documenter suffisamment pour arriver à livrer toutes les fonctionnalités attendues
3.	Manque de compétences d'un ou plusieurs membres de l'équipe sur un aspect du projet	Faible	Moyenne	<ul style="list-style-type: none"> • Mise à niveau par un autre membre de l'équipe, par un camarade de classe ou par le professeur
4.	Risque que le projet soit rejeté	Faible	Élevé	Tout le long du projet communiquer avec le client pour être certain que les deux parties sont alignées

3 Spécification des exigences

3.1 Clarification des exigences du client

3.1.1 Module 1

a) Ajouter des utilisateurs dynamiquement à l'aide d'un fichier csv ou manuellement depuis le terminal

- **Manuellement** : à partir du terminal, on demande tour à tour à l'utilisateur d'entrer les paramètres du compte à créer.

- **Dynamiquement** : on demande à l'utilisateur d'entrer le chemin du fichier .csv et un script crée automatiquement tous les utilisateurs enregistrés dans le fichier
- **Fichier .csv** : fichier structuré dont les éléments y sont séparés par des points virgules
- b) Vérifier l'existence d'un utilisateur**
 - Le nom du compte d'utilisateur (identifiant) ne doit pas être sensible à la casse lors de la vérification
 - Le nom et le prénom par contre doivent prendre la casse en considération
- c) Désactiver et activer les comptes des utilisateurs**
 - Empêcher à un compte de pouvoir se connecter (lorsqu'il est désactivé), ou permettre à un compte désactivé de pouvoir à nouveau se connecter.
 - L'utilisateur ne doit plus avoir accès au système lorsqu'il est désactivé
- d) Supprimer un utilisateur. En cas de suppression d'un utilisateur son dossier principal doit être archivé, conservé et supprimer 30 jours après automatiquement**
 - En créant l'archive, le dossier initial est supprimé

3.1.2 Module 2

- a) archiver les dossiers principaux des utilisateurs dans un dossier partagé :**
 - Créer une archive du dossier personnel de chaque utilisateur et copier cette archive dans un dossier partagé sur un serveur distant.
 - Comparer la signature numérique (hash) de l'archive sur le serveur distant et celle de l'archive locale, si elles diffèrent, recommencer la copie en écrasant la précédente copie. Si les signatures sont les mêmes, supprimer l'archive locale.

3.1.3 Module 3

- a) la criticité des données est très élevée et aucune restriction financière**

Mettre en place des redondances dans au moins une autre zone de disponibilité et dans une autre région.

- b) Les instances de base de données sont privées contrairement aux serveurs d'application**

Mettre en place au moins un sous-réseau privé et un sous-réseau public

c) Les équipements privés sont moins nombreux que ceux qui sont publics

Réserver plusieurs adresses publiques chez AWS pour permettre aux équipements publics d'être disponible sur internet

3.2 Exigences fonctionnelles

3.2.1 Environnement du système

Afin de mieux analyser le programme que nous allons réaliser, nous le découpons en parties nommées « packages » qui représentent des familles de fonctionnalités. Pour les deux premiers modules du projet, ces packages sont :

- **La gestion des utilisateurs** : Ce package regroupe les fonctionnalités d'ajout et de suppression des utilisateurs, la vérification de l'existence d'un utilisateur, l'activation et la désactivation des comptes d'utilisateurs.
- **La gestion des données des utilisateurs** : Ici, on retrouve les opérations qui gère le cycle de vie des données des utilisateurs supprimés et qui s'occupe du backup des données des utilisateurs existant.
- **L'administration** : il s'agit des opérations qui consiste à gérer les droits des utilisateurs vis-à-vis de l'exécution, de la modification et de la lecture du programme.

Le diagramme suivant est une illustration de ces packages.

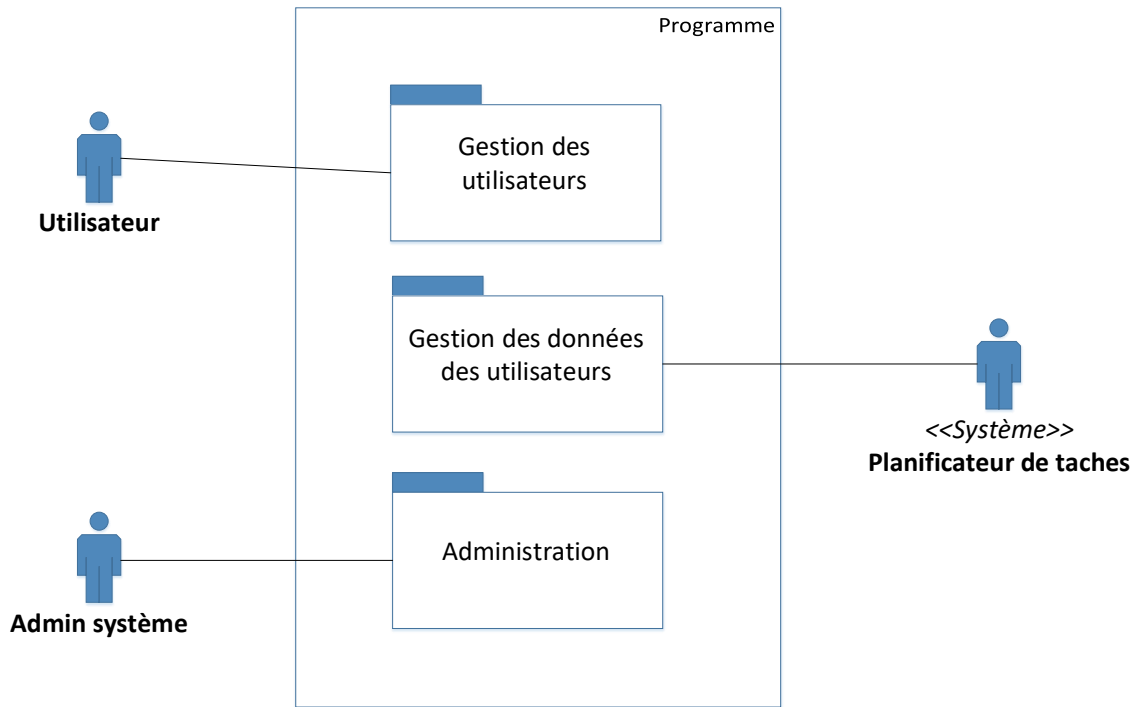


Figure 2 : Diagramme des packages

3.2.2 Les acteurs

Un acteur d'un système est toute entité susceptible d'interagir avec celui-ci. Dans notre cas, les principaux acteurs sont :

- **L'utilisateur** : C'est celui qui utilise le programme bash pour effectuer les tâches d'administration
- **L'administrateur système** : Celui qui donne des droits aux utilisateurs afin qu'ils puissent exécuter le programme
- **Le planificateur** : C'est l'outil système qui permet de lancer automatiquement et de façon régulière l'exécution de certaines actions de notre programme.

3.2.3 Les cas d'utilisation

Un cas d'utilisation est une manière spécifique d'utiliser un système. Un cas d'utilisation réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Nous distinguons les cas d'utilisations suivants :

- **Ajouter des utilisateurs** : c'est la création d'un compte d'utilisateur soit de manière manuelle en entrant tour à tour les paramètres du compte à créer, soit de manière dynamique passant comme paramètre un fichier CSV. L'ajout manuel se

fait suivant une interface à partir du terminal. Avant chaque ajout d'utilisateur, une vérification qu'un utilisateur pareil n'existe pas déjà peu importe la casse est faite.

- **Vérifier l'existence d'un utilisateur** : L'utilisateur entre un nom d'utilisateur et le programme lui dit si l'utilisateur existe déjà ou pas, la vérification se fait peu importe la casse du nom entré en paramètre.
- **Activer/Désactiver un compte d'utilisateur** : Cela consiste pour un utilisateur qui existe au préalable, de lui permettre de se connecter au système ou de lui retirer ce droit.
- **Supprimer un utilisateur** : C'est la suppression d'un compte d'utilisateur existant. Le programme renvoie un message de confirmation si l'opération s'est bien déroulée ou un message d'erreur dans le cas contraire.
- **Archiver le dossier principal de l'utilisateur** : Cela se fait automatiquement après la suppression d'un utilisateur. Cela se fait pour un utilisateur qu'on a supprimé.
- **Supprimer l'archive** : L'archive créée lors de la suppression d'un utilisateur est supprimée automatiquement 30 jours après que l'utilisateur ait été supprimé.
- **Archiver les dossiers** : Archiver automatiquement chaque 7 jours les dossiers principaux des utilisateurs
- **Sauvegarder les archives sur un serveur** : dans un dossier partagé sur un serveur de backup.

3.3 Exigences non fonctionnelles

3.3.1 Modules 1 et 2

- Les scripts/programmés doivent répondre à une problématique à la fois, sinon on se retrouve avec un code trop complexe
- En cas d'entrée erroné de l'utilisateur, le programme doit l'alerter sur l'erreur et proposer un format de réponse
- L'utilisateur doit entrer les informations consécutivement (une à la fois) dans le programme. Chaque information doit être vérifiée

- La sécurité : comparer les signatures des données sauvegardées sur le serveur de backup et celles locales afin de savoir si elles ont été altérées lors du transfert vers le serveur, et si c'est le cas, recommencer la sauvegarde.

3.3.2 Module 3

- Le dessin de l'architecture doit être sobre, lisible et facilement interprétable.

Références bibliographiques