Intitulé de l'épreuve : Conception logicielle

Nombre de copies : 11

Numérotez chaque page (dans le cadre en bas de la page) et placez les feuilles dans le bon sens.

Partie 1

1.1) La solution que l'on réalise à la demande des autorités du MEAE répond au défi de l'augmentation constante du nombre de demandes de rendez-vous dans les différents postes (ambassades et consulats) du réseau diplomatique. L'analyse de besoin de cette application de gestion des rendez-vous permet de détailler les principales fonctionnalités, des deux principaux types d'utilisateurs :

→ l'utilisateur extérieur (usager) peut créer un rendez-vous via l'application et l'annuler.

→ l'usager est alerté lorsque son rendez-vous approche, à temps fixe, par exemple 48 heures avant. Via email, messages etc.

→ l'usager est également alerté si son rendez-vous est annulé ou modifié.

→ les agents consulaires peuvent gérer la disponibilité des services de leur poste, arrêter le service de l'administration des français de l'étranger sur le poste de Pékin par exemple. La prise de rendez-vous par l'usager n'est donc plus possible, et le redémarrer.

→ les agents peuvent modifier les rendez-vous de leur poste et les annuler.

→ les agents disposent de rapports et statistiques sur l'activité de leur poste, via outils de reporting et monitoring (type PowerBI etc)

→ la nature des données étant sensible et selon la réglementation en vigueur, ces données devront être chiffrées et protégées.

→ l'application devra prendre les rendez-vous en fonction des fuseaux horaires de chaque postes.

→ les usagers et agents doivent pouvoir se connecter et s'authentifier, avec différents niveaux d'habilitations pour les agents au besoin.

N° 1 / 11

1.2/ Le premier critère de choix est la sécurité. La nature des données étant sensible, la solution choisie devra impérativement respecter les normes en vigueur en matière de protection des données (RGPD) et les bonnes pratiques recommandées par l'ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information).
Le coût de la solution est également un critère important, ce coût ne doit pas dépasser le budget imposé. La solution doit être maintenable et doit pouvoir subir les différences de montée en charge, que ce soit par rapport à la différence de taille des différents postes ou alors des variations d'affluence, à l'approche de période de vacances par exemple.

1.3/ L'organisation de ce projet s'articule ainsi :
Le MEAE, commanditaire du projet est l'acteur appelé MOA (maîtrise d'ouvrage). La MOE (Maîtrise d'Oeuvre) est l'acteur qui va être chargé d'élaborer le projet, en nommant un chef de projet, à la tête d'une équipe de développement. L'utilisateur final de la solution appelé service métier fera également partie des échanges. L'organisation de ce projet est en agilité (suivant les principes de la méthode AGILE), notamment afin d'éviter l'effet tunnel de développement (commun aux cycles en cascades par exemple). L'utilisateur final est au coeur du projet et des échanges afin qu'à chaque itération, on puisse ajuster les fonctionnalités au besoin. Lors de ces échanges réguliers (toutes les deux semaines par exemples), les différents acteurs échangent sur l'avancement du projet, suivant la feuille de route prévue initialement. L'objectif est d'atteindre au plus vite un MVP (Most Valuable Product), ainsi en cas d'arrêt du projet pour raison budgétaire, on dispose d'un produit utilisable même si partiellant. En méthode SCRUM, un PO (Product Owner) peut également être chargé de faire l'interface entre le besoin métier et les équipes de développements, afin de lever les zones d'ombre et éviter les approximations dans les développements. Enfin, on intégrera nos tests et une démarche devOps à l'exploitation.

1.4/ Afin de planifier le projet, on détaille ses grandes phases:

① Analyse du besoin : étude de faisabilité du projet, analyse du rapport bénéfice / risque, compréhension du besoin

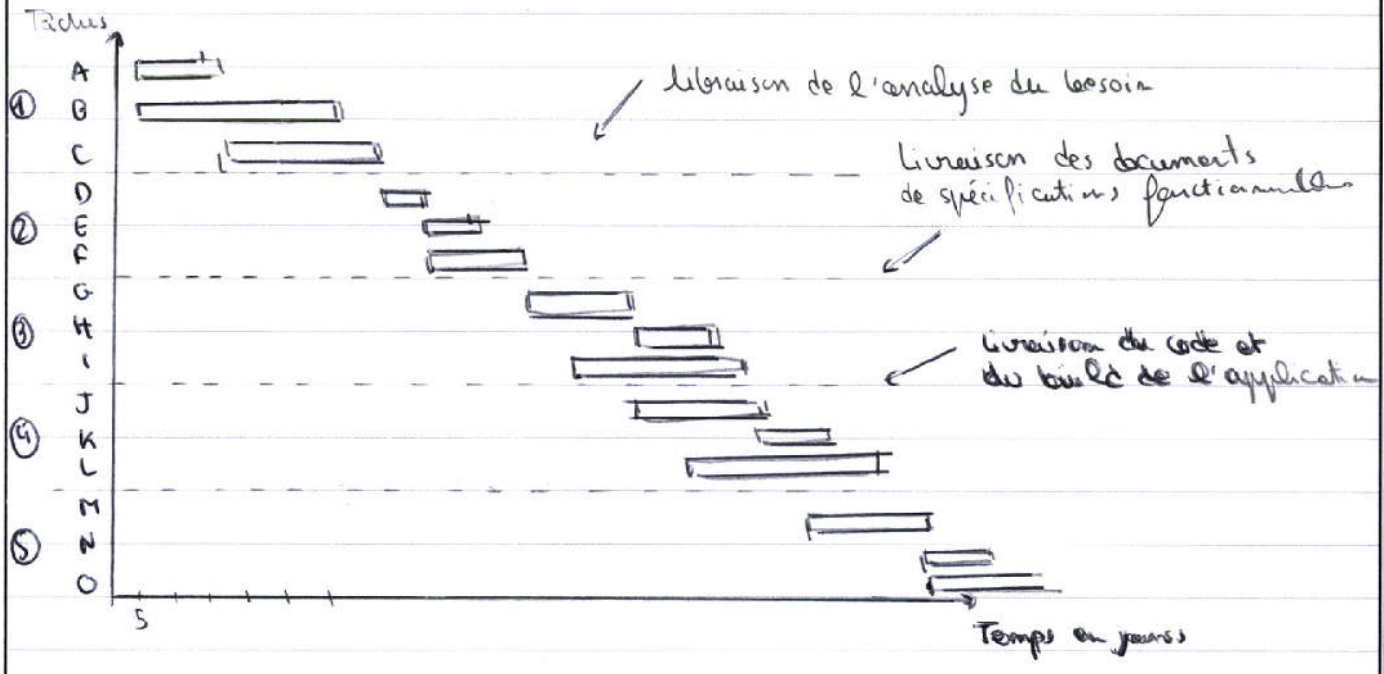
② Spécification fonctionnelle : description fonctionnelle besoin, production des différents diagrammes, diagramme de cas d'utilisation, diagramme de séquence etc

③ Développement : écriture du code par les équipes de développeurs, tests des différentes fonctionnalités.

④ Mise en oeuvre : déploiement en production de la solution, formation aux utilisateurs.

⑤ Exploitation : démarche DevOps, maintenance et développement continus, mise en production automatisée (CI/CD).

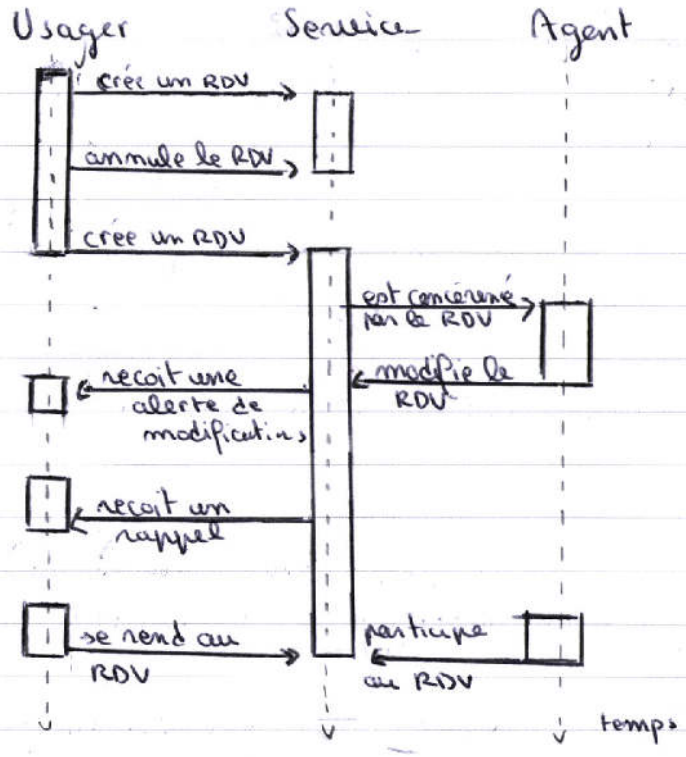
On ces phases et les tâches qui les composent sous la forme d'un diagramme de Gantt :



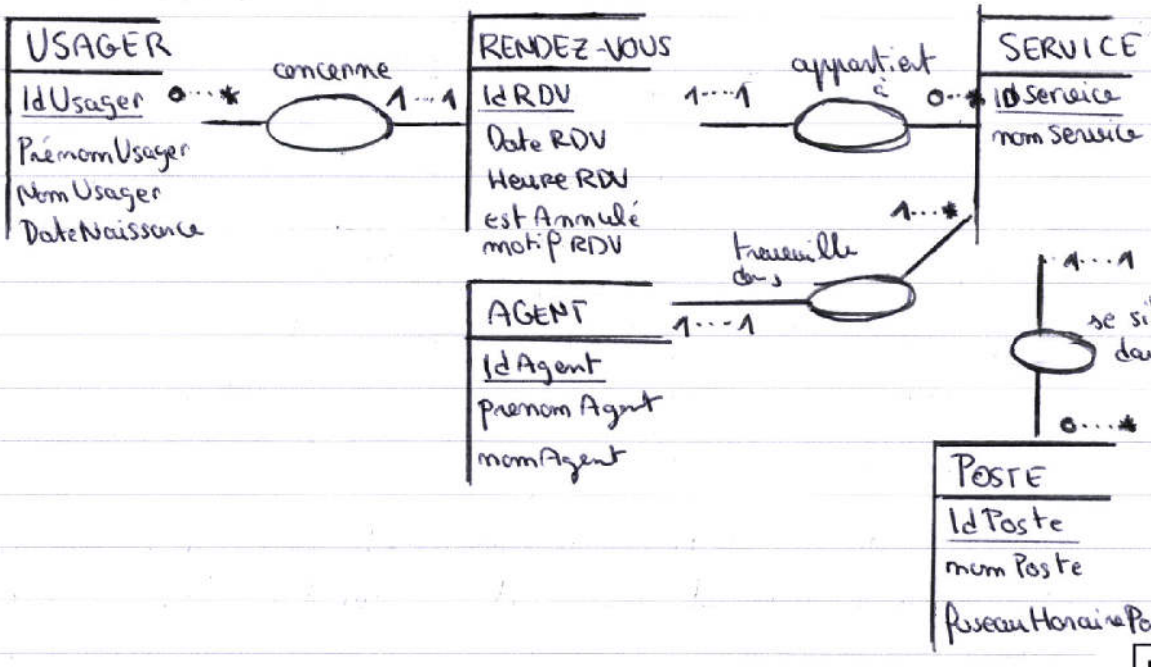
Ce type de représentation permet de faire apparaître le chemin critique du projet et les relations de dépendances entre les tâches.

Partie 2

2.1/ On présente le processus de prise de rendez-vous sous la forme d'un diagramme de séquence :



2.2/ On représente les entités composant cette application par ce diagramme de classe :



Intitulé de l'épreuve : Conception logicielle

Nombre de copies : 11

Numérotez chaque page (dans le cadre en bas de la page) et placez les feuilles dans le bon sens.

suite 2.21, Partie 2 :

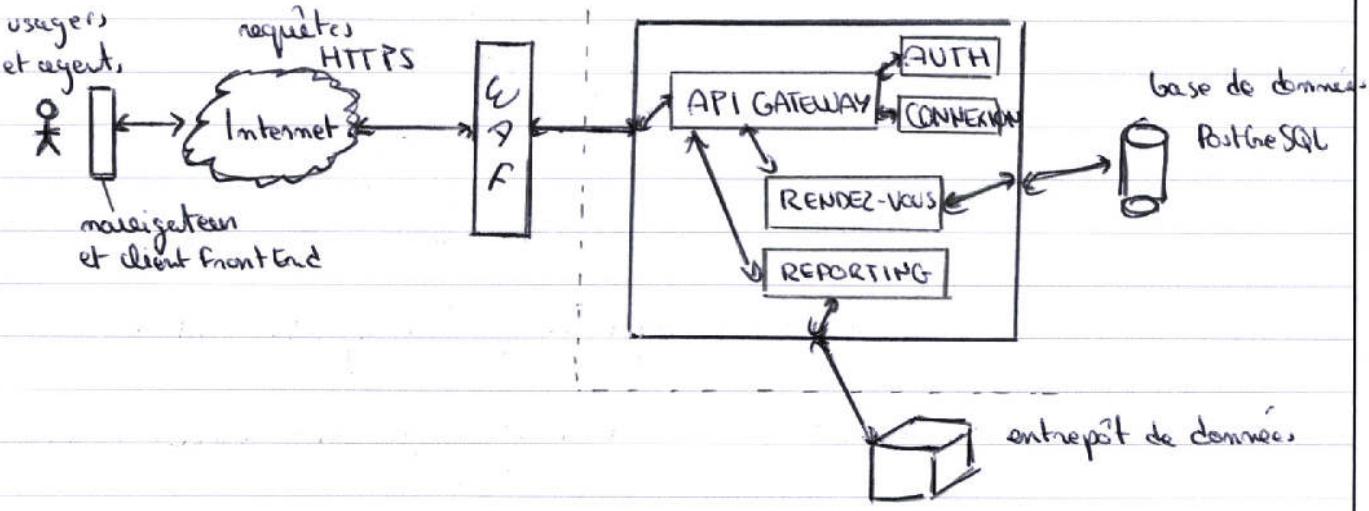
On estime que dans notre système, les agents sont liés à leur service.
L'utilisateur ne voit pas directement (pas directement associé) à un agent.
Un service est lié à un poste, qui contient un fuseau horaire.

Au sujet des cardinalités : un usager a 0 ou de multiple RDV, le RDV est concerné par un seul usager. Le RDV concerne un seul service mais un service peut avoir 0 à de multiples RDV. Un agent travaille dans un seul service mais un service à au moins un agent. Un service concerne un poste mais un poste peut avoir de 0 à de multiple service.

2.31 Ainsi, on choisi de développer cette solution sous la forme d'une application Web, hebergée en Centrale, afin d'être accessible aux agents dans le réseau diplomatique et aux usagers. L'architecture de la solution est dites en microservices, s'opposant aux architectures monolithiques, développée à l'aide du framework Spring, en langage Java. Le client (Front End) de notre application sera sous la forme d'une Single Page Application et developpé avec la librairie React, en langage Javascript. Les données de notre application seront contenues chiffrées dans une base de données PostGreSQL et les données

N° 5/11

nécessaire au reporting et monitoring seront disposées de manière anonymisée dans un Entrepôt de données, sous forme multi dimensionnelle (schéma en étoile). Le serveur hébergeant la solution sera placé dans une DMZ (zone démilitarisé) au sein du réseau, pour sa sécurité. Un WAF (Web application Firewall) est placé entre le serveur et le web.



2.4/ Afin de conceptualiser les interfaces adéquates à l'utilisation de l'application, il convient d'avoir une approche d'UX design.

L'objectif est de rendre l'application confortable à l'usage, avec le minimum de clics dans les différents cas d'utilisation, des couleurs claires, respectant la charte graphique de l'Etat. On peut également effectuer des tests utilisateurs et une analyse qualité afin de vérifier que notre appli est adaptée

Partie 3

3.1/ Les tables sont :

Usager (idUsager , prenomUsager , nomUsager)

RendezVous (idRDV , dateRDV , heureRDV , estAnnuléRDV ,
motifRDV, # idUsager , # idService)

Service (idService , nomService)

avec # sont les clés étrangères

La requête est :

```
SELECT  nomUsager, prenomUsager, motifRDV, heureRDV
FROM    S Service
JOIN    R RendezVous  ON  S.idService = R.idService
JOIN    U Usager  ON  R.idUsager = U.idUsager
WHERE   S.nomService = "Service X"
        AND  R.dateRDV = getDate()
ORDER BY  heureRDV
```

3.2/ On écrit le code demandé sous forme de pseudo code décomposé en deux algorithme :

avec la liste N de RDV ⟶ l = N , on considère l non vide

```
Algo TauxAnnulation ( l listeRDV ) ⟶ listeRDV
    nbrRDVannulé = 0 ;  nouvelle liste        // cet algo renverra la liste
    nbrRDV = taille(l) ;  R ;                    des RDV annulés
    Tant que l non vide faire                 // on parcourt la liste
        si estAnnulé ( l.tête ) alors         // on teste la tête de la liste
            nbrRDVannulé + + 1 ;
            ajoute ( l.tête , R ) ;           // on ajoute la tête à la liste
        fin si                                   que l'on va renvoyé
    fin tant que
    affiche " le taux d'annulation de RDV est : "
            + nbrRDVannulé / nbrRDV
    retourne R
```

N° ... / ...

```
Algo TauxAnnulationRDVparAgentetUsager (l listeRDV) → vide
    nouvelle liste la = TauxAnnulation(l);        // on appelle notre premier algo, affichant le taux
    nbrRDVannulé = taille(la);
    nbrRDVannuléUsager = 0;
    nbrRDVannuléAgent = 0;
    Tant que la non vide faire :
        si annuléUsager(tête.la) alors            // on teste la tête, on considère qu'on dispose des algos pour le faire
            nbrRDVannuléUsager ++ 1
        sinon
            nbrRDVannuléAgent ++ 1
        fin si
    fin tant que
    Afficher " taux d'annulation par usager : " + nbrRDVannulUsager / nbrRDVannulé
            + "taux d'annulation par agent :" + nbrRDVannuléAgent / nbrRDVannulé
fin algo
```

3.3/ La bonne accessibilité de l'application consiste à respecter les pratiques préconisées par la RGAA, permettant au maximum d'utilisateur de travailler sur l'application, notamment le personnel et les usagers atteint de handicap.

3.4/ Il convient d'effectuer des tests réguliers afin de vérifier la qualité de l'application, notamment avec une démarche de tests type test unitaire, test de non regression, vérifier que notre application ne subit pas d'effet de bord.

Intitulé de l'épreuve : Conception logicielle
Nombre de copies : 11

Numérotez chaque page (dans le cadre en bas de la page) et placez les feuilles dans le bon sens.

Partie 4

4.1/ Dans le cadre de ce type d'hébergement, 3 types de Cloud sont possible, fourni par un prestataire extérieur, le fournisseur cloud :

Iaas, Infrastructure as a Service, le fournisseur cloud fourni les ressources informatiques nécessaire, serveur, réseau etc. sur son site / datacenter.
PaaS, Platform as a Service, le fournisseur cloud fourni en plus des ressources informatiques, le système d'exploitation, eventuellement les machines virtuelles.

SaaS, Software as a Service, le fournisseur fourni complétement la solution en l'hébergeant, la maintenant etc. On accède alors à l'appli fourni clé en mains par requête HTTP.

Le problème de l'utilisation de service cloud public est la perte de souveraineté sur la donnée. Un meilleur contrôle est possible via cloud privé / hybride.

4.2/ Sur ce type d'application, différentes types d'attaques sont à craindre. Tout d'abord l'attaque par déni de service (DDOS), on s'en prémunit grâce à parefeu rejetant les IP cherchant à surcharger les services. Les attaques par injection SQL sont aussi à redouter. Il est nécessaire de tester les entrées utilisateurs au sein de notre application (les formulaires par exemple) afin de s'assurer que notre utilisateur

n'injecte pas de code qui pourrait altérer voir détruire notre base de données. Des attaques par ingénierie sociale peuvent également survenir. Il convient de former les utilisateurs aux bonnes pratiques et à la gestion des mots de passe et accès.
Une bonne gestion des risques de sécurité lors du développement de l'application est nécessaire afin de la protéger. En cas de survenance, les équipes d'exploitation doivent être alertés et réagir vite afin de protéger la solution et ses données.

Partie 5

5.1/ Les termes MCO et MCS sont respectivement Maintien en condition opérationnelle et Maintien en condition de sécurité

Cela consiste en l'ensemble de méthode et de pratique visant à garder la solution en bon état de fonctionnement et dans un état protégé.

5.2/ ainsi, 3 demandes d'évolutions sont transmises :

① afin de permettre à un usager, représentant un groupe ou une famille, de pouvoir grouper les RDV, il faudra effectuer une aggregation de l'objet RDV de notre solution. Ainsi, un RDV pourra ou non faire partie d'un groupe de RDV. Le champ nouveau IdGroupe permettra

de caractériser celui (si il est nul, le RDV n'a pas de groupe)
Il n'est pas nécessaire que notre nouvelle classe Groupe soit lié à l'usager, on récupère l'Usager via les RDV et on n'implémentera l'impossibilité d'attribué des RDV ayant un usager différent, dans un même groupe.

② Afin que l'application soit disponible en anglais et arabe, il faudra implémenter une traduction en fonction de la zone géographique de l'usager. Cette modification s'effectue au niveau du client, donc sur le Front End de l'application.

③ Afin de répondre à cette demande, il conviendrait d'intégrer à notre solution des API, interface de programmation applicative. Ces nouvelles briques logicielles pourront présenter via URI des ressources comme le planning des services ou la liste des RDV. Ces briques logicielles pourront donc être intégrées et interrogées par d'autres applications du ministère.
On interroge ces API via requête HTTP (à l'aide de nombreuses méthodes comme GET, PUT, POST etc).

Par exemple, une requête GET sur l'URI "\tous les RDV" permettra à une autre application d'obtenir les RDV sous format JSON.

N°

... /...