

 Avenue V. Maistriau 8a
B-7000 Mons
 +32 (0)65 33 81 54
 scitech-mons@heh.be

WWW.HEH.BE

Projet interdisciplinaire

Cahier des charges

Bachelier en Informatique – Orientation réseaux &
télécommunications – Bloc 2

Erwin Desmet
erwin.desmet@heh.be

Antoine Malaise
antoine.malaise@heh.be

Yoan Pietrzak
yoan.pietrzak@heh.be

Johan Depréter
johan.depreter@heh.be

Table des matières

1.	Description du projet	2
1.1	Comportement	2
1.2	Groupe	2
1.3	Autonomie	3
1.4	Evaluation	3
2.	Objectifs généraux	4
3.	Infrastructure réseau & Systèmes	5
3.1	Configuration réseau	5
3.2	Serveurs	5
4.	Développement des applications	6
4.1	Applications locales (PHP)	6
4.2	Applications globales (Python ou PHP)	6
4.3	Bonnes pratiques	6
5.	Analyse	7
6.	Notions de sécurité de l'AD et journalisation de la base de données	7
7.	Documentation technique	8
8.	Planning suggéré	8
9.	Responsabilités par département	9
10.	Cahier des charges par département	9

1. Description du projet

Le projet interdisciplinaire a pour objectif de vous immerger dans une simulation professionnelle qui allie infrastructure réseau, systèmes informatiques, bases de données, développement web, et gestion collaborative. Chaque groupe représentera une entité indépendante (par exemple, un hôpital) avec des responsabilités propres, tout en collaborant avec les autres pour contribuer à un système centralisé interconnecté. Il est évident que dans un tel projet, toutes les technos ne sont pas connues et qu'un apport de votre part et d'un partage de connaissance dans votre équipe est plus qu'important.

Je rappelle qu'il s'agira d'un travail de groupe entre les deux options, et vous devrez donc partager vos connaissances et vous coordonner selon les méthodes de votre choix.

La semaine de projet aura lieu la dernière semaine du quadrimestre 1 durant la fin du mois de décembre.

Ce projet est structuré autour de trois dimensions clés :

1. **Technique** : Création d'une infrastructure réseau, mise en place de serveurs, développement web en PHP/Python, mise en place d'un système de sécurité basé sur l'AD.
2. **Collaborative** : Intégration de données partagées dans un système global sécurisé, interactions inter-groupes.
3. **Documentaire** : Fourniture d'analyses UML, modélisation de la DB, de maquettes, de plans réseau, et de guides utilisateurs.

1.1 Comportement

Durant cette semaine de projet, vous devrez faire preuve de professionnalisme. Et donc trouver une méthodologie de travail de groupe adéquate, une cohésion, un échange professionnel entre vous et les professeurs servant d'expert ou ceux servant de client. L'interprétation des informations sera différente selon l'un et l'autre. Il sera donc important d'employer un vocabulaire adéquat en fonction.

Chaque membre du groupe devra participer au projet et se voir attribuer des tâches ! Votre travail au sein de l'équipe sera évalué durant la semaine celui qui n'aura pas participé de manière active à l'élaboration pourra se voir exclu du projet et donc se verra attribué la note de 0/20.

1.2 Groupe

L'élaboration du projet se fera comme convenu au départ par groupe de 5 étudiants maximum. Chaque groupe devra comporter plusieurs profils tel que :

- **1 Chef d'équipe** : Il sera la personne ressource pour le client et sera donc l'UNIQUE référent pour celui-ci. Bien entendu il fera partie de l'équipe de développement/réseau ce qui veut dire qu'il devra participer de manière active à l'élaboration du projet.

- **2 développeurs** : Ces personnes auront une spécificité beaucoup plus orienté développement. Ils seront chargés d'élaborer le développement des différents modules.
- **2 gestionnaires réseau** : Ces personnes auront plusieurs missions d'aider les développeurs en testant le code et de gérer l'infrastructure réseau.

Ces rôles sont théoriques, simplement pour que les professeurs aient une personne de contact par point de travail, par exemple si un problème en Linux arrive, une aide sera demandée par un des gestionnaires réseau.

1.3 Autonomie

Durant cette semaine de projet, vous devrez faire preuve d'autonomie, ce critère entrera dans votre évaluation.

Il existe énormément de ressources sur les concepts du projet. Il vous est donc demandé de réaliser des recherches personnelles et approfondies d'un point de vue programmation. Référez-vous par exemple avec Django, flask pour la partie Python ou les framework tels que Laravel ou Symfony pour le PHP ... si vous désirez utiliser des frameworks ou bibliothèques en partie programmation.

Les enseignants joueront plusieurs rôles :

- **Le client** : vers qui vous devrez vous tourner pour valider vos avancements, maquettes, ...
- **L'expert** : vers qui vous pourrez aller lorsque vous aurez des questions d'ordre technique.
- **Vous ne pouvez les contacter que sur votre équipe en taguant Expert** : votre question ou **Client** : votre demande.

1.4 Evaluation

L'évaluation sera réalisée à différents niveaux :

- **Présentation du produit final aux clients** : le vendredi 20 décembre à partir de 13h00 vous devrez présenter aux différents clients votre projet. Cette présentation se fera sous forme de démonstration. A ce moment tous les profs sont des clients. L'ordre de passage pour cette après-midi, vous sera donné par la suite.
- **Le code** : Chaque groupe devra déposer sur Moodle (dans la rubrique Dépôt - Code source) une copie de son code le vendredi 20 décembre après-midi avant 16h30.
- **Configuration réseau et infrastructure** : Vos protocoles de configurations, vos scripts, votre plan d'adressage, votre topologie ... devront être déposés avec votre code.
- **La rédaction du rapport** : Chaque groupe devra rédiger un rapport de maximum 25 pages. Vous trouverez le canevas de rapport dans les notes de cours de Gestion de projet. Le rapport devra comprendre :
 - Une page de garde
 - Une table des matières
 - Une introduction
 - Une présentation générale du projet

- Une présentation détaillée des composants
- Une description des **algorithmes** de programmation et de vos configurations sur vos serveurs/réseaux
- Les problèmes rencontrés
- Les améliorations et conclusions
- Bibliographie
- Les annexes (si vous en avez)

Le rapport devra être déposé **sur Moodle** (dans la rubrique Dépôt – Rapport ou sur la boîte mail suivante : erwin.desmet@heh.be) **le vendredi 20 décembre avant 16h30.**

- **La documentation demandée dans le cahier de charges** : Ces documents devront être déposés sur Moodle (dans la rubrique Dépôt – Documentation) **le vendredi 20 décembre avant 16h30**
- Une copie de tous vos dépôts peut être envoyée à erwin.Desmet@heh.be pour sécurité.

ATTENTION : Une remise tardive des différents documents, le non-respect des types de fichiers ou le non-respect des emplacements de dépôts sera considéré comme non remis et aura pour résultat un 0 pour la partie en question.

2. Objectifs généraux

Le projet vise à développer une simulation de gestion interconnectée pour des hôpitaux. Chaque groupe représentera un hôpital distinct, avec des spécialités et des responsabilités propres. Les différents hôpitaux (Hôpital Général, Hôpital Pédiatrique, Soins Intensifs, Réhabilitation) doivent collaborer pour partager les données critiques (lits disponibles, transferts de patients, équipements critiques) via une plateforme centralisée.

- **Infrastructure réseau et systèmes** :
 - Configurer des VLAN et des serveurs DNS/DHCP.
 - Une Vlan par département.
 - Gérer des bases de données locales et une base de données centrale interconnectées.
 - Installer et gérer les serveurs Windows/Linux pour les besoins spécifiques et communs.
- **Développement d'applications web** :
 - Applications locales en PHP pour gérer les besoins spécifiques de chaque groupe (à chaque département).
 - Une application globale en Python ou PHP centralisant les données partagées.
 - Mise en place de contrôles d'accès via AD et vérification des permissions selon les groupes.

- **Documentation complète et précise :**
 - Analyses UML (diagrammes de cas d'utilisation, classes, séquences).
 - Modélisation de la base de données.
 - Maquettes des interfaces utilisateur.
 - Rapports détaillés sur les tests et les configurations techniques.

3. Infrastructure réseau & Systèmes

3.1 Configuration réseau

Chaque classe disposera d'un switch configuré avec les VLAN suivants :

- **VLAN 10, 20, 30, 40** : Attribués aux groupes 1 à 4. Chaque VLAN est isolé pour des tâches spécifiques.
- **VLAN 100** : VLAN commun regroupant les services partagés (DNS principal, base centrale, application commune).

VLAN 10 (Hôpital Général) : Admissions classiques et statistiques d'occupation.

VLAN 20 (Hôpital Pédiatrique) : Chambres adaptées et suivi des patients par âge.

VLAN 30 (Soins Intensifs) : Lits critiques et gestion des équipements d'urgence.

VLAN 40 (Réhabilitation) : Séjours longue durée et équipements de rééducation.

VLAN 100 (Commun) : Hébergement de la base centrale et des services partagés.

Routage entre VLANs :

- Activé sur le switch pour permettre aux VLAN locaux de communiquer avec le VLAN commun uniquement.

3.2 Serveurs

Chaque groupe disposera de serveurs locaux et collaborera avec un serveur commun.

- **Serveur Windows Local (par VLAN)**
 - **DNS secondaire** : Celui-ci devra correspondre à la zone de votre département.
 - **DHCP failover** : Fournir des adresses IP aux appareils du VLAN en cas de panne du DHCP principal.
- **Serveur Windows commun (VLAN 100)**
 - **Active Directory (AD)**
 - Gestion des utilisateurs globaux.
 - Validation des connexions par adresse IP et VLAN.
 - Contrôle d'accès pour les applications globales.
 - **DHCP Principal pour les 4 VLANS**
 - **DNS Principal** : Devra contenir une zone Hopital.lan, celle-ci contiendra une zone par département (ex : Pédiatrie.Hopital.lan).
- **Serveur Linux Local (par VLAN)**
 - **Serveur Web (Apache/Nginx)** : Hébergement des applications locales (PHP).

- **Base de données locale** : Gestion des données spécifiques au groupe (patients, équipements). On y stockera notamment les lits disponibles, les patients, ...
- **Serveur Linux Commun (VLAN 100)** :
 - **Base de données centrale** : Hébergement des tables partagées (lits disponibles, transferts).
 - **Application globale (Python/PHP)** : Centralisation des données et services accessibles à tous.

4. Développement des applications

4.1 Applications locales (PHP)

Chaque groupe doit :

1. Développer une **interface CRUD** pour gérer les données locales (ex. lits, équipements).
2. Assurer la synchronisation de certaines données locales avec la base de données centrale.
3. Implémenter des fonctionnalités spécifiques liées à leurs responsabilités (statistiques, rapports).
4. Implémentations d'une architecture MVC
5. Fonctionnalités spécifiques :
 - **Hôpital Général** : Gestion des patients, des lits, ...
 - **Hôpital Pédiatrique** : Gestion des chambres adaptées, visualisation des rapports pédiatriques, ...
 - **Soins Intensifs** : Gestion des lits d'urgence, visualisation des statistiques, ...
 - **Réhabilitation** : Gestion des séjours, gestion des équipements, ...

4.2 Applications globales (Python ou PHP)

Cette application doit :

1. **Consolider les données** : Afficher le flux du trafic, les alertes, etc. Ainsi que présenter un historique complet des transactions.
2. **Gérer les alertes** : Notifications en temps réel.
3. **Fournir un tableau de bord général** : Récapitulatif des capteurs/alertes.
4. **Assurer la sécurité des données** : Validation par IP, vérifier que l'utilisateur ait une IP dans le bon VLAN et permissions SQL. Authentification par AD. Sécurisation des requêtes SQL

4.3 Bonnes pratiques

Cette application doit :

- Répondre aux normes de bonnes pratiques vu en cours
- Être documentées
- Être sécurisé

- Documenter de la BONNE manière
- Ergonomique
- Responsive

Chaque groupe doit appliquer les méthodologies et utiliser les outils vus en cours pour améliorer la cohésion et l'échange avec le client et les experts au cours du projet.

5. Analyse

Chaque groupe doit produire les diagrammes UML suivants :

1. **Diagrammes de cas d'utilisation :**
 - Exemple : Gestion d'un transfert entre deux hôpitaux.
2. **Diagrammes de classes :**
 - Exemple : Lit, Chambre, Patient, ...
3. **Diagrammes de séquence :**
 - Exemple : Flux d'une demande de lit depuis une application locale vers l'application centrale.

Chaque groupe doit produire les modélisations des bases de données.

Chaque groupe doit produire le plan réseau : VLAN, Serveurs, configuration IP, ...

Chaque groupe doit produire un plan de sauvegarde.

Chaque groupe doit prévoir un plan de Testing.

6. Notions de sécurité de l'AD et journalisation de la base de données

Authentification :

Tous les utilisateurs doivent s'authentifier via l'Active Directory (local ou central) avant d'accéder aux applications.

Toutes les données sur l'AD doivent être sauvegardées et toutes les mesures de sécurités vues au cours doivent être mise en place.

Contrôle d'accès :

Permissions basées sur les groupes AD (ex. un utilisateur du Groupe 1 ne peut pas accéder aux ressources du Groupe 2).

Validation des actions critiques (modifications, transferts) selon le rôle défini dans AD.

Optionnel - Traçabilité :

Journaux de connexion enregistrés sur le serveur AD pour chaque utilisateur. Via un script powershell par exemple ou autre méthode. Suivi des modifications dans les bases de données via triggers.

7. Documentation technique

Chaque groupe doit livrer une documentation complète comprenant :

1. **Plans réseau** : VLAN, serveurs, et configurations IP.
2. **Maquettes** : Interfaces des applications locales et globale.
3. **Scripts de configuration** : Fichiers de configuration pour les serveurs Linux/Windows.
4. **Rapports de tests** : Résultats des tests unitaires et d'intégration.

8. Planning suggéré

Jour 1 : Analyse et Planification

- Étude des besoins spécifiques et communs.
- Création des diagrammes UML, modélisation de la base de données et des maquettes.

Jour 2 : Configuration Réseau et Serveurs

- Configuration des VLAN.
- Installation et configuration des serveurs (DNS, DHCP, Web, Bases de données).

Jour 3 : Développement Local (PHP)

- Implémentation des fonctionnalités CRUD locales.
- Tests unitaires sur les bases de données locales.

Jour 4 : Application Commune (Python ou PHP)

- Développement des fonctionnalités globales.
- Tests d'intégration entre bases de données locales et centrale.

Jour 5 : Validation et Présentation

- Validation réseau, serveurs, et applications.
- Présentation des résultats par chaque groupe.

9. Responsabilités par département

Hôpital Général (VLAN 10) :

- Gestion des admissions classiques.
- Suivi des statistiques d'occupation des lits.
- Gestion des catégories de lits (standard, soins prolongés).

Hôpital Pédiatrique (VLAN 20) :

- Gestion des chambres adaptées pour enfants (par tranche d'âge).
- Suivi des rapports sur les admissions pédiatriques.
- Notification des pénuries de lits critiques.

Soins Intensifs (VLAN 30) :

- Gestion des lits d'urgence (tri par disponibilité immédiate).
- Gestion des équipements critiques (respirateurs, moniteurs).
- Tableau de bord pour les patients en soins intensifs.

Réhabilitation (VLAN 40) :

- Gestion des séjours longue durée (dates d'entrée et sortie).
- Suivi des équipements de rééducation (fauteuils roulants, équipements sportifs).
- Gestion des priorités pour les transferts entre hôpitaux.

10. Cahier des charges par département

1. Chaque département est responsable de ses données locales et de leur synchronisation avec la base de données centrale.
2. Les applications locales doivent inclure une interface simple, ergonomique et intuitive (HTML/CSS/JavaScript).
3. Les permissions d'accès doivent être gérées via Active Directory et des validations côté serveur (PHP/Python).