



Cahier des charges

1. Contexte et présentation du projet

1.1. Objet du document

Ce document contient le cahier des charges de la seconde version de l'application GSB compte rendu.

1.2. Présentation de l'entreprise

Le laboratoire Galaxy Swiss Bourdin (GSB) est issu de la fusion entre le géant américain Galaxy et le conglomérat européen Swiss Bourdin.

En 2009, les deux géants pharmaceutiques ont uni leurs forces pour créer un leader de ce secteur industriel. L'entité Galaxy Swiss Bourdin Europe a établi son siège administratif à Paris.

Le siège social de la multinationale est situé à Philadelphie, Pennsylvanie, aux Etats-Unis.

La France a été choisie comme témoin pour l'amélioration du suivi de l'activité de visite.

1.3. Définition du besoin

1.3.1. Définition de l'objet

L'activité commerciale d'un laboratoire pharmaceutique est principalement réalisée par les **visiteurs médicaux**.

Les **visiteurs médicaux** (ou *délégués*) démarchent les médecins, pharmaciens, infirmières et autres métiers de santé susceptibles de prescrire aux patients les produits du laboratoire.

L'objectif d'une visite est d'actualiser et rafraîchir la connaissance des professionnels de santé sur les produits de l'entreprise. Les visiteurs ne font pas de vente, mais leurs interventions ont un impact certain sur la prescription de la pharmacopée du laboratoire.

Le laboratoire souhaite développer une application de gestion des visites. **Il désire mettre à disposition des visiteurs médicaux des pages Web permettant de centraliser les comptes-rendus de visite.** Cette base d'information sera utilisée à des fins d'élaboration de la démarche de communication auprès des praticiens et donnera une vision individuelle et synthétique de l'activité de représentation.

Pour permettre une aide au renseignement des rapports, l'application fournira une description des produits du laboratoire, les coordonnées précises des praticiens et des informations détaillées les concernant.

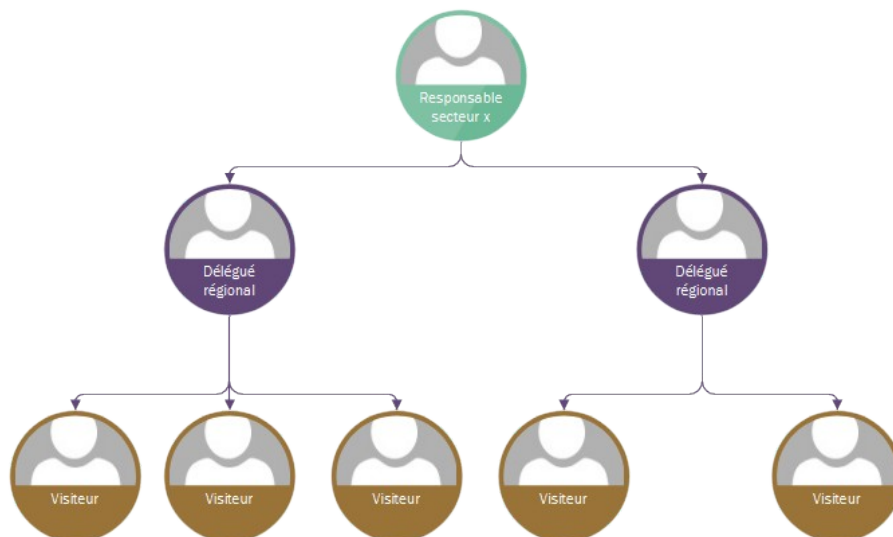
Elle servira aussi à la mise en relation de la hiérarchie de la force commerciale, des visiteurs aux responsables de secteur en passant par les délégués régionaux.

1.3.2. Forme de l'objet

Sous forme de pages Web sécurisé et authentifié, l'application pourra devenir dans le futur un outil accessible depuis des terminaux mobiles (tablettes ou téléphones).

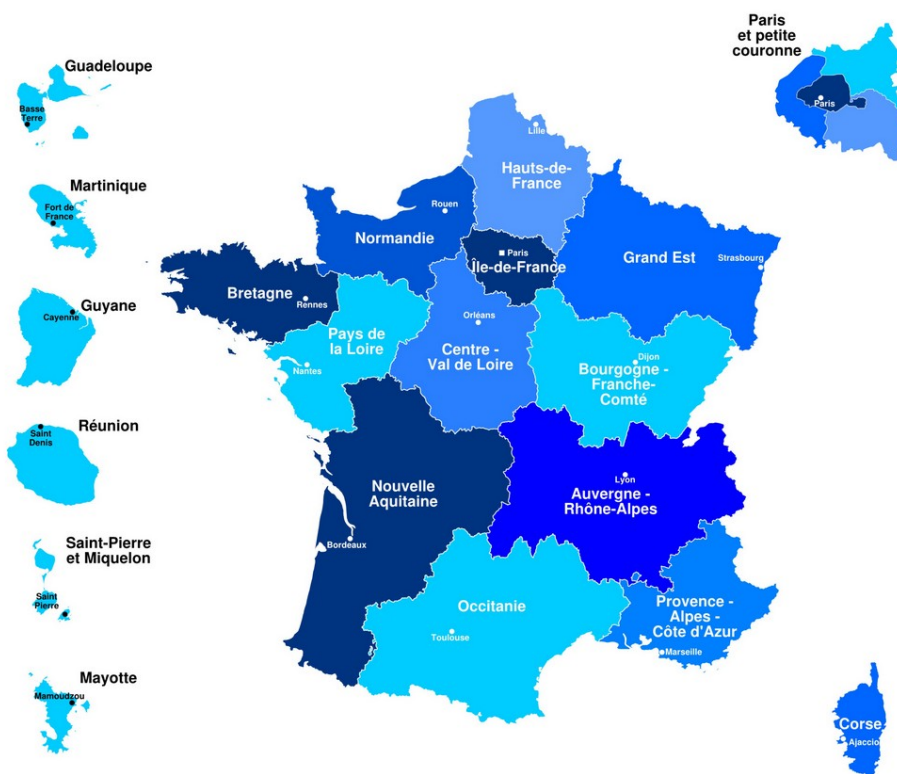
1.3.3. Organisation de force commerciale

Pour donner une organisation commune aux délégués médicaux, l'entreprise a adopté l'organisation de la flotte de **visiteurs** selon un système hiérarchique par région : le **délégué régional** et à un niveau supérieur par **secteur géographique** (Sud, Nord, Paris-Centre, Antilles-Guyane, etc) : **Responsable de secteur**.



la

Carte des régions de France



<https://www.welcometofrance.com/>

1.3.4. Les produits

Les produits distribués par le laboratoire sont des **médicaments** : ils sont identifiés par un numéro de produit (dépôt légal) qui correspond à un nom commercial (ce nom étant utilisé par les visiteurs et les médecins).

Comme tout médicament, un produit a des effets thérapeutiques et des contre-indications.

Un produit relève d'une **famille** (antihistaminique, antidépresseur, antibiotique, ...).

Lors d'une visite auprès d'un médecin, un visiteur présente un ou plusieurs produits pour lesquels il pourra laisser des échantillons. De manière à pouvoir gérer les coûts des visites, le prix de l'échantillon est une donnée à retenir dans la base.

1.4. Description de l'existant

En 2003, le laboratoire Swiss Bourdin avait mené un travail d'informatisation des comptes-rendus, qui devait permettre aux visiteurs de disposer d'un outil de gestion personnelle de leur activité et, par un module d'export, d'adresser aux délégués régionaux une synthèse mensuelle.

L'application était prévue sous Access et certaines parties ont été développées.

Le cahier des charges de l'application Access **Version2003 est disponible**.

Ce cahier des charges comprends le MCD version 2003 et une documentation utilisateur.

On souhaite repartir de l'étude de 2003 et développer une application de gestion des comptes rendus par des pages Web qui assureront la gestion des comptes-rendus.

Des entretiens menés en début d'année 2023 actualisent le contexte depuis l'analyse de 2003 et constituent le recueil des besoins.

2. Description graphique et ergonomique

L'outil Adobe XD version gratuite pourra être utilisé
<https://www.adobe.com/fr/products/xd.html>

2.1. Charte graphique :

Le design des pages devra être en conformité avec l'ébauche proposée
Exemple d'ébauche de page :

La couleur prédominante doit être le bleu. Le logo ne doit jamais changer de couleur.
(Pour les couleurs de votre charte graphique: <http://colorshemesdesigner.com/csd-3.5/#>)

2.2. Design de la page :

Chaque page doit être composée d'un menu. Le menu doit comprendre un accès à toutes les pages disponibles sur le site selon l'authentification de l'utilisateur.

Exemple de saisie d'un rapport de visite :

2.3. Ergonomie

Saisie du rapport N°1

Formulaire permettant de rédiger un rapport de visite.

*Champs obligatoires

Rapport de visite

Numéro du rapport : 1	Date de visite * : 01 / 12 / 2022
Matricule du collaborateur : a131	Motif * : Périodicité
Praticien concerné * : Gosselin Albert	1er médicament présenté: DEPRAMIL
Remplaçant Marçais-Lefebvre Bertrand	<input checked="" type="checkbox"/> Echantillon
Date de saisie * : 14 / 12 / 2022	
Bilan du rapport * : Voici le bilan magnifique du rapport ptdr	AMOPIL 1 +
<input type="radio"/> Saisie définitive	
Valider le rapport	Retour

On limitera tous les éléments graphiques superflus (pas de menu animé...). On pourra améliorer ou contrôler l'interface utilisateur par du code Javascript.

2.4. Arborescence du site :

Une maquette de l'arborescence du site fera l'objet d'une validation par les utilisateurs.

3. Description fonctionnelle et technique

3.1. Description fonctionnelle :

Les cas d'utilisations sont décrits dans le fichier joint à ce cahier des charges.
De plus le recueil des besoins a été mené et est disponible dans le fichier joint.

3.2. Contraintes techniques :

3.2.1. Environnement

L'application Web sera développée dans un langage de script côté serveur, éventuellement complétée par des API, frameworks et autres compléments.

Les langages de programmation retenus pour ce projet sont le PHP, HTML, JavaScript. Le site doit être compatible avec Firefox, Chrome et Edge.

Dans le cadre d'une portabilité vers un environnement d'application mobile, on prendra soin de séparer au maximum le rendu utilisateur de la logique interne de l'application.

3.2.2. Architecture

L'architecture du code produit doit respecter les conventions d'usage : développement basé sur une architecture **MVC**.

3.2.3. Modules

1. Un premier module **Comptes-rendus** différent selon le profil de l'utilisateur :
 - Visiteur : permet la saisie des comptes-rendus, la consultation des données sur les trois années précédentes, une vision synthétique du travail réalisé (visites sur une période donnée, statistiques diverses, etc.). **C'est le module le plus urgent et le seul décrit ici.**
 - Délégué : *permet la vision de l'activité des visiteurs rattachés à une région, sous forme de statistiques, graphiques ou par une vision de l'activité de chaque employé. Il propose aussi la saisie de compte-rendu de visite (puisque les délégués restent acteurs sur le terrain).*
 - Responsable de secteur : *permet la vision de l'activité des visiteurs rattachés à un secteur sous forme de statistiques diverses, de graphiques, pour un visiteur en particulier ou pour l'équipe dépendant d'une région.*
2. un module de **consultation**
 - des visiteurs
 - des médicaments
 - des praticiens

Documentation

La documentation devra présenter l'arborescence des pages pour chaque module, le descriptif des éléments classes et bibliothèques utilisées, ou autres éléments externes utilisés.

3.2.4. Base de données :

La base de données de l'application access 2003 devra être migrée vers une base de données MariaDB afin de permettre la migration des données dans son intégralité. Cette base de données devra être modifiée pour répondre aux nouvelles exigences fonctionnelles. Les données sont modélisées avec un MCD conformément au besoin de la solution applicative.

Les accès aux données sont contrôlés conformément aux habilitations définies par le cahier des charges.

La base de données est **sauvegardée** selon la planification à définir et des tests de restauration sont à effectuer.

La base de données est opérationnelle et stable dans l'environnement de production.

3.2.5. Tests fonctionnels

Les **tests fonctionnels** sont à décrire et à réalisés. Une documentation est fournie à cet effet les décrits dans le fichier joint à ce cahier des charges.

Principe :

- le développeur écrit le tableau des tests fonctionnels pour la USR développée. Les valeurs à tester sont indiquées.
- Lorsque la Usr passe de « done » à « on test », le binôme suit le tableau des tests fonctionnel prévu et valide ou pas le développement en indiquant le résultat obtenu dans le tableau.

3.2.6. Gestion de versions

un outil collaboratif de gestion des itérations de développement et de versions est utilisé ;

- une documentation des versions vient appuyer l'intégration continue ;

Voir cours de Madame Jouin : gestion de version Git + power point

<https://www.youtube.com/watch?v=hPfgekYUKgk>

4. Prestations attendues et méthodologie de suivi

4.1. Prestations attendues :

Les attendus de cette application sont la réalisation des pages décrites et conformes aux cas d'utilisation, du design des pages validées par les utilisateurs et à une documentation technique.

Les livrables attendus pour l'examen comprennent :

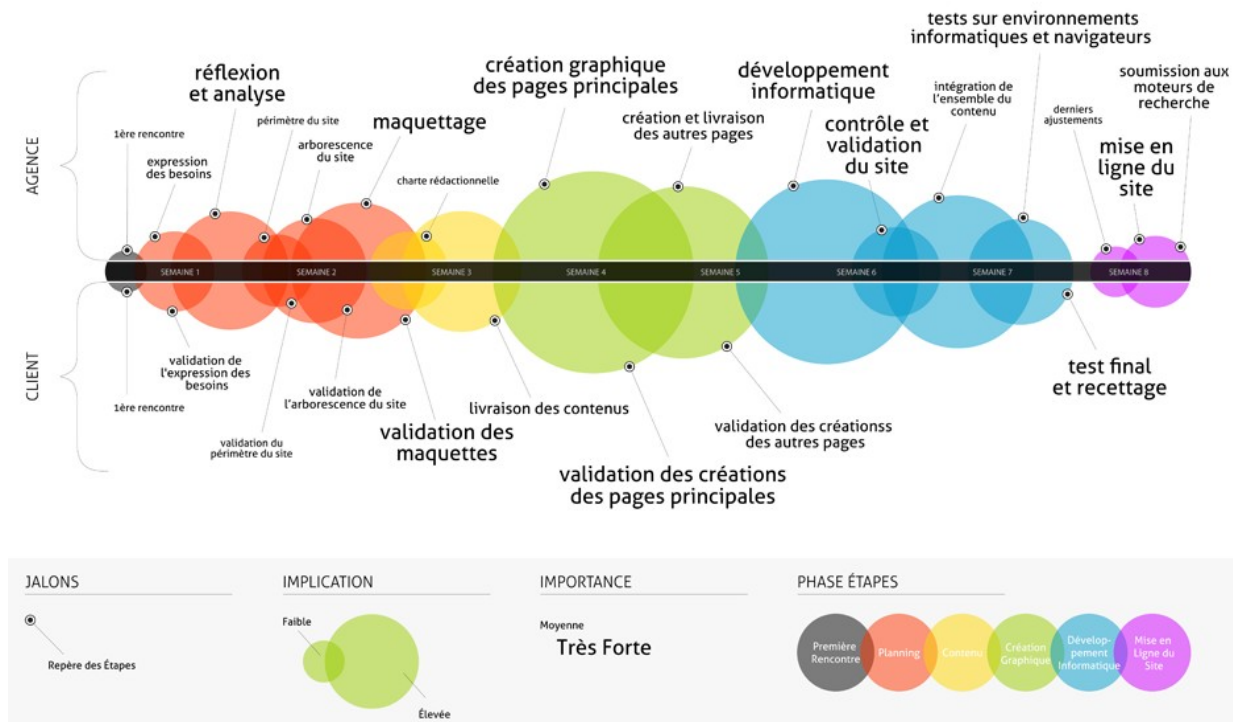
- Les codes sources du site
- **Une documentation technique (Voir TP Mme Jouin Doxygen) et utilisateur**

L'application doit être fonctionnelle sur le serveur de production.

4.2/ Planning

La date limite est fixée au 11 décembre

Le planning respectera la chaîne de production d'un site suivante :



Sources : <http://www.webdesign2803.fr/infographie/la-chaîne-de-production-dun-site-internet/attachment/a-website-designed1/>

Le projet sera découpé en différentes phases :

- 1.Phase de prototypage : Reflexion et analyse, Construction des maquettes, Définition des tests fonctionnels
- 2.Phase de développement du pilote : Développement de l'ensemble des fonctionnalités, intégration des maquettes, contrôle et validation du site, tests
- 3.Phase de mise en production sur servslam20.

4.3/ Méthodologie

L'ensemble du projet sera basé sur une méthodologie agile type "**scrum**".

Attribution des rôles :

Product Owner : Un membre de l'équipe de GSB (Mme Méret). Il sera en charge de définir les priorités et de prendre les décisions lors de potentiels arbitrages.

Scrum Master : Un membre de l'équipe du prestataire (Mme Méret) qui sera en charge du respect de la méthodologie et des relations avec le Product Owner.

-Construction du backlog. Pour toutes les phases, un product backlog découpant l'ensemble des fonctionnalités et éléments à réaliser sera défini.

-Sprints : Le projet sera organisé sur la forme de sprints mensuels. Ces sprints se décomposeront toujours de la manière suivante :

Etape 0 –L'équipe fixe l'objectif général du sprint à partir du Product Backlog.

Etape 1 –L'équipe définit la liste des tâches à réaliser au cours du sprint. Chaque tâche est évaluée en temps afin d'ajuster le nombre de tâche à réaliser en fonction de la capacité de développement disponible sur le sprint.

Etape Fin :

L'équipe définit l'objectif principal du prochain sprint. (à faire valider par le Scrum Master)

Démo – l'équipe du prestataire présente au Product Owner les résultats du travail réalisé au cours du dernier sprint. La démo se fera sur un environnement de pré-production
L'équipe définit la liste des tâches du prochain sprint (Sprint Backlog)

5. Sécurité – voir cours cybersécurité web

La gestion des différentes failles de sécurité qui pourront être amené à arriver avec un site web et une base de données.

5.1. Faille include et les failles sur les variables de sessions

Les failles sur les includes ne doivent pas être exploitables par de tierce personne. Les failles sur les variables de sessions ne doivent pas être permises pour cela vous devrez faire des vérifications sur les `user_id` sauvegardez et faire les comparaisons nécessaires avec les données en BDD.

1.1. Injection SQL

Pour lutter contre les injections SQL, on utilisera la fonction `mysql_real_escape_string` pour empêcher les caractères spéciaux contenus dans les chaînes de caractères entrées par l'utilisateur. On pourra également utiliser des requêtes préparées pour empêcher un éventuel code inséré dans les paramètres d'être interprété.

1.2. Faille XSS et CSRF

Il faudra également prévoir les contre-mesures pour lutter contre les failles XSS en convertissant les données via des fonctions utilisées dans le langage que nous avons choisi, le PHP. Pour ce qui est des contre-mesures contre le CSRF on utilisera des confirmations à l'utilisateur pour les actions.