- Rapport de projet -

Création d’un système de suivi des commandes jusqu’au déploiement

Chef de projet : ABEMONTY Timothée
ARMOURDOM Suvan
AMIRY Ben
ANDRIAMANANTSOA Rochi
HOARAU Thomas

BUT Réseau et Télécommunication – 1ère année

Tuteur : HOARAU Cédric

IUT de la Reunion

Table des matières

1. INTRODUCTION

La tendance actuelle des outils numériques, tant dans leur évolution technologique que dans leur intégration aux processus métiers, représente une avancée majeure dans leur utilisation professionnelle.
Illustré par la montée en puissance des solutions SaaS, les logiciels de gestion et de suivi de commande sont des exemples concrets de ces évolutions. Ils permettent aux entreprises de mieux organiser leurs flux logistiques, de suivre en temps réel les commandes clients, et d'optimiser la communication entre les différents services impliqués (logistique, production, service client). Des plateformes telles que Odoo, Shopify ou encore Prestashop montrent l'ampleur de ce phénomène, aussi bien pour les petites entreprises que pour les grandes structures cherchant à digitaliser leur gestion commerciale.

Nous avons donc choisi, pour notre projet tutoré, de développer notre propre logiciel de suivi de commande, en nous inspirant des fonctionnalités proposées par les solutions existantes sur le marché.
L'objectif est de concevoir une application web intuitive permettant aux utilisateurs de créer, suivre et gérer les commandes tout au long du processus de traitement, depuis la validation jusqu'à la livraison finale. Nous mettons à disposition des outils simples mais essentiels pour assurer une transparence totale sur l'état des commandes, tout en facilitant la collaboration entre les intervenants. Nous ne cherchons pas à intégrer toutes les fonctionnalités possibles dès le départ, mais à répondre de manière efficace aux besoins principaux d'un suivi opérationnel.

Pour mener à bien ce projet, nous avons commencé par recenser et analyser les besoins fonctionnels à travers une phase d'expression des exigences. Cette étape nous a permis de définir les grandes lignes du logiciel et de cibler les fonctionnalités prioritaires. Par la suite, nous avons mené plusieurs phases de tests afin de confronter les résultats obtenus aux attentes initiales, d'identifier les écarts et d'améliorer notre application. Enfin, nous avons dressé un bilan complet de notre démarche et des résultats obtenus.

2. DESCRIPTION DU PROJET

- **Authentification**

Pour accéder à l'interface de suivi et gérer les commandes, l'utilisateur doit s'authentifier à l'aide de son identifiant et de son mot de passe. Cette étape est nécessaire pour garantir la sécurité et la confidentialité des données.

- **Tableau de bord**

L'utilisateur dispose d'un tableau de bord personnalisé, qui affiche un résumé des activités : commandes en cours, livraisons prévues, alertes sur les retards ou les anomalies. Il peut également y consulter l'historique de ses actions.

- **Gestion des commandes**

L'utilisateur peut créer, visualiser, modifier ou annuler une commande selon ses droits. Chaque commande contient les informations suivantes : numéro, date, client, produits, statut (en attente, validée, en cours, livrée), et commentaires éventuels.

- **Suivi en temps réel**

Le logiciel permet de suivre en temps réel l'évolution de chaque commande : de sa création jusqu'à la livraison. Des notifications automatiques peuvent être envoyées à différentes étapes (validation, expédition, retard…).

- **Recherche de commandes**

Un outil de recherche permet de filtrer les commandes selon différents critères : numéro de commande, client, statut, date, ou encore type de produit. Cela facilite l'accès rapide à une commande spécifique.

- **Liste de clients**

Chaque utilisateur peut consulter la liste des clients enregistrés dans la base. Il peut également ajouter un nouveau client ou modifier ses informations (adresse, contact, conditions de livraison…).

- **Visibilité et confidentialité**

Chaque utilisateur peut définir le niveau de visibilité de ses informations ou de celles des commandes dont il est responsable. Il peut limiter l'accès à certaines données uniquement à son équipe ou à des membres spécifiques.

3. RÉALISATION TECHNIQUE

3.1 Environnement et technologies utilisées

Dans le cadre du développement de notre logiciel de suivi de commande, nous avons mis en place une machine virtuelle Debian via VirtualBox. Ce choix nous a permis de travailler dans un environnement isolé, stable et proche des conditions d'un serveur réel. Nous y avons installé un serveur web (Apache2), une base de données MySQL et PHP pour gérer la logique côté serveur.

Cet environnement nous a permis de centraliser notre développement, de tester l'application en local et de garantir une bonne compatibilité entre les différents composants.

Langages et outils utilisés

- **HTML**

Le langage HTML a été utilisé pour structurer les différentes pages de l'interface utilisateur. Chaque page représente une fonctionnalité spécifique du logiciel : création de commande, tableau de suivi, fiche client, etc.

- **CSS**

La présentation graphique du site a été gérée à l'aide de feuilles de style CSS, ce qui nous a permis d'alléger le code HTML et de centraliser la gestion de l'apparence de l'application. Cette séparation du fond et de la forme a facilité les modifications visuelles et l'uniformité du design sur l'ensemble du site.

Exemples d'utilisation :

- Personnalisation des tableaux de commandes
- Mise en forme conditionnelle selon l'état des commandes
- Adaptation de l'interface selon le type d'utilisateur connecté

- **PHP**

Côté serveur, nous avons utilisé PHP pour assurer la logique applicative. PHP a servi à traiter les formulaires, gérer les connexions utilisateurs, interagir avec la base de données MySQL et sécuriser l'application avec des sessions.

Exemples d'utilisation :

- Enregistrement des commandes

- Gestion de l’authentification
- Cryptage des mots de passe
- Traitement des formulaires (ajout, modification, suppression)

- **MySQL**

Nous avons utilisé MySQL pour stocker l’ensemble des données de l’application : utilisateurs, produits, commandes, livraisons, etc. Ce système de gestion de base de données relationnelle est bien adapté pour des applications web grâce à sa performance, sa stabilité et sa compatibilité avec PHP.

- **JavaScript**

Le JavaScript a été utilisé côté client pour rendre l’interface plus interactive. Il permet notamment de valider certains formulaires avant envoi, de gérer des événements utilisateurs (clics, sélections), et d’améliorer l’expérience générale sans recharger les pages.

3.2 Les tests

Afin d'assurer la qualité, la stabilité et la conformité de notre application, nous avons mis en œuvre une stratégie de tests rigoureuse tout au long du développement, réalisée sur une machine virtuelle Debian (via VirtualBox). L'application étant développée avec PHP, MySQL, HTML/CSS et JavaScript, il était essentiel de tester chaque composant de manière ciblée.

Nous avons ainsi conduit plusieurs types de tests :

- Tests unitaires
- Tests d’intégration
- Tests fonctionnels
- Tests de performance
- Tests d’utilisabilité

- Tests unitaires

Les tests unitaires visent à vérifier le bon fonctionnement individuel de chaque fonctionnalité, indépendamment du reste de l’application. Ils ont été menés lors du développement de modules tels que :

- L’inscription utilisateur
- La connexion
- La création de commande
- L’ajout de produit

Chaque script PHP a été testé pour vérifier que les entrées étaient correctement traitées, les erreurs détectées, et que les sorties correspondent aux attentes définies dans le cahier des charges. Une fois validés, les fichiers étaient transmis pour intégration.

- Tests d’intégration

Les tests d’intégration nous ont permis d’assembler les différents modules validés unitairement, pour s'assurer de leur bonne compatibilité entre eux. Ce type de test est crucial pour détecter les erreurs d’interconnexion, souvent invisibles lors des tests isolés.

Exemples de vérifications réalisées :

- Compatibilité des sessions PHP entre les modules
- Absence de redondance de variables globales
- Cohérence des échanges entre les formulaires (HTML) et leur traitement côté serveur (PHP/MySQL)

Des problèmes récurrents ont été relevés, notamment des conflits de sessions ou des appels à des fonctions incompatibles, rapidement corrigés après analyse.

- Tests fonctionnels

Les tests fonctionnels valident que les fonctionnalités du site sont bien conformes aux exigences définies. Ils consistent à simuler l’utilisation réelle du site, en vérifiant que :

- Chaque page répond correctement aux actions de l’utilisateur
- Les cas d’erreurs (mots de passe incorrects, champs vides, données incohérentes) sont correctement gérés
- Les messages d’erreur s’affichent de manière claire

Par exemple, lors du test du module "messagerie", nous avons détecté que la suppression d'un email ne fonctionnait pas comme prévu, ce qui a nécessité une correction du code PHP.

- Tests de performance

Deux vagues de tests ont été réalisées :

1. Avant optimisation, avec des requêtes SQL

2. Après simplification du code PHP et des requêtes

Résultat : le temps d'affichage a été réduit de moitié, montrant l'impact significatif d'une optimisation ciblée.

- Tests d'utilisabilité

Enfin, des tests d'utilisabilité ont été réalisés auprès d'un panel de 4 utilisateurs, représentatifs de notre cible. Ces tests avaient pour objectif de :

- Identifier les zones de confusion ou de friction

- Synthèse

Les tests réalisés tout au long du développement, dans notre environnement virtuel sous Debian, ont permis de garantir la fiabilité de notre application web. Ils ont été essentiels pour :

- Identifier les bugs techniques

- Vérifier la conformité avec le cahier des charges

- Améliorer l'expérience utilisateur

Ce processus itératif de test, correction et validation nous a permis de livrer un outil stable, rapide et ergonomique.

4. Organisation et bilan du projet

4.1 Organisation du groupe : méthode de travail

Dès le lancement du projet, nous avons défini une méthode de travail collaborative en nous appuyant sur les compétences de chaque membre de l'équipe. La répartition initiale se basait sur les technologies dominées par chacun (frontend, backend, design, etc.), mais cette organisation a rapidement évolué, notamment en raison du travail à distance.

Nous avons alors adopté une organisation afin d'assurer une coordination et un suivi des tâches. Chaque membre avait des responsabilités définies par fonctionnalité, et le travail s'effectuait en autonomie avec des points réguliers pour synchroniser les avancées.

Exemple de processus de développement d'un module :

1. Spécification technique : les tâches étaient ensuite décomposées en éléments frontend (maquettes, CSS/JS) et backend (API, base de données).

2. Développement individuel : chaque élément était développé de manière indépendante en suivant une convention de nommage et de structure définie pour assurer la compatibilité lors de l'intégration.

3. Tests unitaires : une fois développée, chaque fonctionnalité faisait l'objet de tests automatisés ou manuels pour valider son bon fonctionnement.

4. Tests d'intégration : enfin, des tests étaient effectués pour vérifier que les modules interagissent correctement ensemble, tant sur le plan fonctionnel que visuel.

Cette organisation nous a permis de gagner en efficacité tout en assurant une qualité de code stable et maintenable.

Avantages :

- Tests et validations systématiques à chaque étape

- Développement en parallèle possible

Inconvénients :

- Risques de conflits de code si les conventions ne sont pas respectées

- Dépendance à la rigueur de chacun pour maintenir la cohérence du projet

Les facteurs clés de succès étaient la clarté dans les échanges, la rigueur dans l'organisation du code, et une communication continue sur l'état d'avancement.

4.2 Résultats obtenus

Les objectifs définis dans le cahier des charges ont globalement été atteints. La majorité des fonctionnalités principales ont été mises en place et sont pleinement opérationnelles.

Parmi les livrables finalisés :

- Authentification et gestion des comptes
- Création, affichage et suppression de contenus
- Interface responsive et intuitive
- Gestion des erreurs et validation des formulaires
- Tests de performance et de montée en charge validés

Nous avons néanmoins fait le choix de concentrer nos efforts sur la qualité et la stabilité du site, en privilégiant les tests, la sécurité, et l'optimisation des performances..

4.3 Améliorations envisagées

Bien que la version actuelle du projet soit opérationnelle, plusieurs pistes d'amélioration ont été identifiées pour des évolutions futures :

- Corriger et enrichir le module de messagerie : assurer la suppression effective des messages, et intégrer des options supplémentaires (marquage, recherche, tri).
- Mise en ligne publique : en configurant un hébergement web adapté, avec déploiement continu via GitHub Actions. Un espace de discussion communautaire (forum ou Discord) pourrait aussi être ouvert pour recueillir les retours des premiers utilisateurs.

5. Conclusion

Ce projet tutoré s'est révélé bien plus ambitieux et complexe que les projets réalisés auparavant au cours de notre formation, tels que les mini-projets. Il a constitué une véritable mise en situation professionnelle, en mobilisant plusieurs compétences acquises tout au long du cursus.

Il nous a permis de mettre en pratique de manière concrète des notions essentielles telles que :

- la modélisation (cas d'utilisation),
- les langages de développement (frontend, backend),
- la gestion de projet (répartition des tâches, respect des deadlines),
- mais aussi des compétences transversales comme la communication, la collaboration et l'adaptabilité.

Chacun des membres de l'équipe a non seulement consolidé ses connaissances, mais a également contribué activement au bon déroulement du projet en partageant son savoir-faire. Cette dynamique collective a été un véritable moteur pour l'efficacité du groupe.

Au-delà de l'aspect technique, ce projet nous a confrontés aux réalités du travail en équipe et à des situations proches de celles que nous rencontrerons en entreprise :

- écouter et intégrer les idées de chacun,
- argumenter pour défendre ses choix techniques,
- s'organiser à distance,
- gérer les imprévus tout en maintenant la cohérence et le rythme global du projet.

Enfin, ce projet nous a appris à gérer un projet dans sa globalité, de la conception initiale à la livraison, en passant par les tests, la documentation et les compromis à faire en cas de contraintes de temps ou de ressources.

En somme, cette expérience a été à la fois technique, humaine et formatrice. Elle nous a offert une première approche concrète de la gestion d'un projet informatique dans un contexte semi-professionnel, et constitue une préparation précieuse pour notre futur stage et notre insertion dans le monde du travail.