Réalisation en milieu professionnel

SIO SLAM 2021-2023

Application web de gestion de ticketing

Une application de gestion de ticketing est un logiciel ou une plateforme en ligne qui permet à une entreprise ou une organisation de gérer les demandes de support ou les problèmes signalés par les clients ou les utilisateurs. L'outil permet de suivre les tickets de support depuis leur création jusqu'à leur résolution, en fournissant une traçabilité complète des interactions avec les clients.

L'objectif principal de ce type d’application est d'améliorer l'efficacité du service clientèle et de fournir une expérience client de haute qualité en résolvant les problèmes rapidement et efficacement.

L’entreprise peut également suivre les tendances et les problèmes récurrents pour identifier les opportunités d'amélioration de leurs produits et services.

Le but de ce projet est de réaliser une version simplifier d’un applicatif de ticketing. Cette application web responsive doit permettre de gérer les conversations avec les clients lors d’un bug ou d’une panne sur un site.

Voici les fonctionnalités que l’applicatif devra pouvoir réaliser :

1. Réception des demandes : Il permet aux utilisateurs de soumettre des tickets de support ou des demandes de service via l’applicatif.
2. Tri et catégorisation : Les tickets sont triés automatiquement ou manuellement en fonction de leur urgence, de leur type ou de leur catégorie pour faciliter leur traitement.
3. Attribution des tickets : Les tickets sont assignés aux membres de l'équipe de support ou de service compétents en fonction de leurs compétences, de leur charge de travail ou d'autres critères définis.
4. Suivi et priorisation : Chaque ticket devra être suivi tout au long de son cycle de vie, afin de connaître son état, de le mettre à jour et de le prioriser en fonction de l'urgence ou de l'importance.
5. Communication et collaboration : L'applicatif de gestion de ticketing facilite la communication entre les utilisateurs et les membres de l'équipe de support. Les commentaires, les mises à jour et les notifications sont souvent intégrés pour maintenir toutes les parties informées de l'avancement du ticket.
6. Base de connaissances : Il peut inclure une base de connaissances ou une bibliothèque de résolution de problèmes qui permet aux utilisateurs de trouver des réponses à leurs questions courantes ou de résoudre des problèmes mineurs sans avoir besoin de créer un nouveau ticket.

Expression des besoins

L’outil de gestion de ticketing permet aux équipes d’un service client de répondre aux demandes et aux réclamations rapidement et précisément.

Voici le parcours d'un ticket de service client étape par étape :

1. Authentification de l’utilisateur

Un utilisateur se connecte à l’outil de ticketing et accède à son interface de gestion de tickets.

2. Création du ticket

Un utilisateur crée un ticket via l’interface web. Il va pouvoir gérer la hiérarchisation des priorités en fonction des demandes et des problèmes des clients. On pourra classer et assigner les tickets en fonction du niveau de priorité et du type de la demande (système, BDD, web, réseau, présentation, back office ….)

3. Activation du service client

L'équipe du service client est avertie de la création d'un nouveau ticket. Un membre de l'équipe s'attribue le ticket, ou le ticket lui est attribué automatiquement, selon le mode de contact utilisé par le client.

4. Résolution du ticket et interaction avec l'équipe

Le ticket est mis à jour par le membre de l'équipe du service client en fonction des informations fournies par le client. Le service en charge du ticket échange avec le client sur l’avancement de la résolution du problème.

5. Fermeture du ticket

Une fois la question ou la demande résolue, la personne chargée du ticket doit le fermer. Cependant, toutes les informations recueillies doivent être conservées dans l’outil de ticketing afin d'être utilisées lors des prochaines demandes ou interactions.

Cahier des charges

La première étape consiste à développer une interface web en PHP et relié à une base de données MySQL qui permettra :

* De se connecter avec un login et un mot de passe à l’interface de gestion des tickets
* D’afficher un tableau permettant de consulter la liste des tickets (voir capture 1)
* De créer de nouveaux tickets avec comme choix le Type de la demande (Système, BDD, Web,…) et le niveau de priorité (normal, urgent et critique) (voir capture 2)

Spécifications fonctionnelles

Vous trouverez dans cette section les spécifications fonctionnelles du projet.

L’application doit être entièrement responsive web design.

Il est conseillé d’utiliser le framework Bootstrap et Boostrap Table pour la gestion des tableaux :

https://getbootstrap.com/

https://bootstrap-table.com/

**BX0 – Gestion des utilisateurs via un compte ADMIN**

**Description :** Page accessible uniquement à l’utilisateur qui possède le rôle ADMIN. Elle devra présenter la liste des utilisateurs avec la possibilité d’ajouter ou de supprimer un accès. L’interface de création de compte doit permettre à un utilisateur se créer un compte avec une adresse mail et un mot de passe. Lors du processus de création, il faudra enregistrer la date de création. Par défaut un accès est désactivé lors de sa création. Seul l’ADMIN peut rendre actif un compte de type USER ou CLIENT.

La table ‘utilisateurs’ contient un seul compte avec le rôle ADMIN. Les autres rôles possibles sont :

USER : collaborateur en charge du traitement d’un ticket

CLIENT : client qui émet un ticket

**Data** : utilisateurs [Login, Mot de passe, Rôle, Actif, Date création]

**BX1 – Page d’accès à l’applicatif de gestion des tickets**

**Description** : Page présentant un formulaire de connexion à l’applicatif via un login et mot de passe. Le mot de passe devra être vérifié en utilisant tous les moyens sécuritaires mis à disposition avec PHP. Si l’utilisateur possède le rôle USER, il accède à la liste des tickets clients à traiter. Il peut alors choisir de traiter le ticket qui lui semble le plus prioritaire.

Si l’utilisateur possède le rôle CLIENT, il accède à l’interface lui présentant la liste des tickets qu’il a créés. Il peut également : créer un nouveau ticket, mettre à jour ou clôturer un ticket.

**Data** : clients [Num client, Id Utilisateur, Nom, Entreprise, Observation], utilisateurs [Login, Mot de passe, Rôle, Actif, Date création]

**BX2 – Tableau d’affichage des tickets**

**Description** : Cette fonctionnalité doit permettre d’afficher la liste des tickets en cours de traitement. La colonne « Numéro demande » permettra d’accéder au détail de la demande BX3 (voir capture 1).

**Data** : tickets [Numéro demande, Type demande, Priorité, Sujet, Message, Pièce jointe]

**BX3 – Création d’un nouveau ticket**

**Description** : Cette fonctionnalité doit permettre de créer un nouveau ticket. Lors de la création du ticket, l’application génère un numéro de ticket qui devra être unique (voir capture 2). Son état sera par défaut ’ouvert’.

**Data** : tickets [Numéro demande, Type demande, Priorité, Sujet, Message, Pièce jointe, Etat]

**BX4 – Affichage du détail d’un ticket**

**Description** : Cette fonctionnalité doit permettre de visualiser le contenu complet d’un ticket avec l’historique des échanges avec le client. L’ordre d’affichage se fera du plus récent vers le plus ancien. En bas de page sera placée une zone

de type « textearea » permettant au client de poursuivre les échanges si le ticket est toujours ouvert. Un bouton placer en haut à droite permettre de clôturer le ticket par le client (voir capture 3).

Chaque ticket clôturé doit être gardé en base de données afin d'être utilisée lors des prochaines demandes ou interactions.

**Data** : tickets [Numéro demande, Type demande, Priorité, Sujet, Message, Pièce jointe, Etat], traitements [Réponse, date réponse]

Base de données :

Les tables suivantes sont obligatoires dans la construction du projet :

* Une table ‘clients’ : [‘Numéro client’, ‘Id utilisateur’,’Nom’,’Entreprise’]\*
* Une table ‘tickets’ : [‘Id Type demande’, ‘Id utilisateur’, ‘Numéro demande’, ‘Id priorité’, ‘Sujet’, ‘Message’, ‘pj’, ‘Etat’]\*
* Une table ‘traitements’ : [‘Id tickets’, ‘Réponse’, ‘Date réponse’]\*
* Un table ‘typedemande’ : [‘Type demande’ (Valeurs exemples : système, BDD, web, réseau, présentation, back office)]\*
* Une table ‘priorite’ : [‘Type priorité (Valeurs exemples : Normal, Urgent, Critique)]\*
* Une table ‘utilisateurs’ : [‘Login, ‘Mot de passe’, ‘rôle’( ADMIN, USER, CLIENT ), ‘Actif’, ‘Date création’]\*

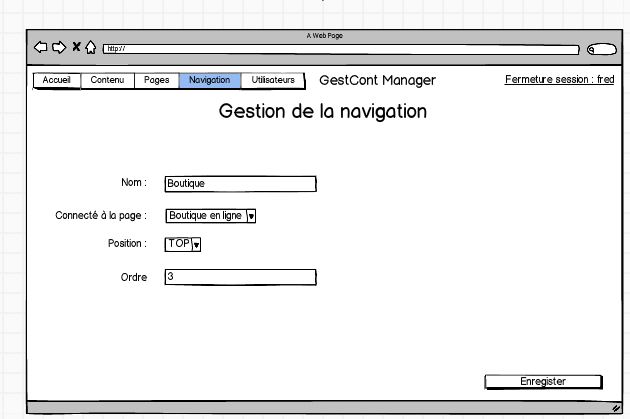
\* Exemple de nom de champs pour les tables données à titre indicatif

D’autres tables pourront être ajoutées en fonction des choix ou des besoins nécessaires identifiés par l’étudiant. Il est demandé de fournir les documents de type Modèle Relationnel et Modèle logique permettant d’établir les liaisons entre chaque table.

Maquettage :

Vous devrez réaliser avec l’outil de votre choix des schémas fonctionnels de chaque pages (desktop et responsive ) et de ses déclinaisons.

Exemple de schémas fonctionnel non représentatif du projet :





Travail à fournir

Au travers de leur espace de travail (serveur web) les étudiants devront organiser les répertoires des différents

contenus (images, feuille de style, javascript, etc..). Prévoir également un répertoire pour y placer les différentes notices et documentations.

Accès à votre espace de développement : http://51.178.86.117:<numero\_de\_port>

Accès à la base de données : <http://51.178.86.117:8087/phpmyadmin/>

Les étudiants devront fournir en plus des divers développements demandés, une notice technique (présentation des scripts, modèle relationnel des données, etc..) et fonctionnel (schéma des divers écrans, parcours de navigation,…)

Résumer du travail à réaliser :

1. Réunion avant projet sur la répartition et l’organisation du travail de chaque partie,
2. Mise en place de l’environnement de travail (Jira, Git, développement),
3. Création du MCD et MLD et de la Base de données,
4. Création des dessins fonctionnels des écrans (mockup) du back-office avec un outil de type Balsamiq Mockups (voir exemple d’écran ci-dessus),
5. Développement de l’interface back-office en proposant un design de type responsive en utilisant un framework de type Bootstrap et Bootstrap Table,
6. Développer l’applicatif en utilisant une architecture MVC. Il est interdit d'utiliser des modèles de template existant. Il est par contre recommandé d'utiliser les frameworks et bibliothèques suivantes :

- Bootstrap et Bootstrap table

- JQuery

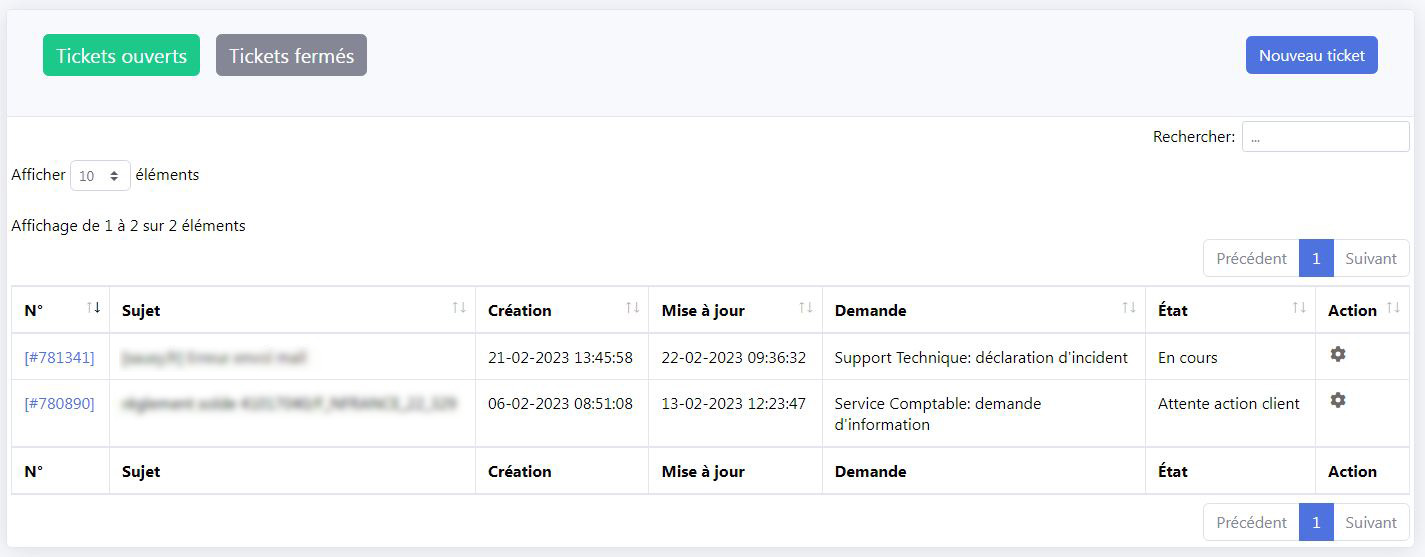
Toutes les étapes du projet devront être implémentées avec la méthode agile en utilisant l’outil Jira Scrum.

La phase de développement devra se faire via l'outil de versioning GIT puis déployer sur GitHub avec au minimum les branches "master", "develop".

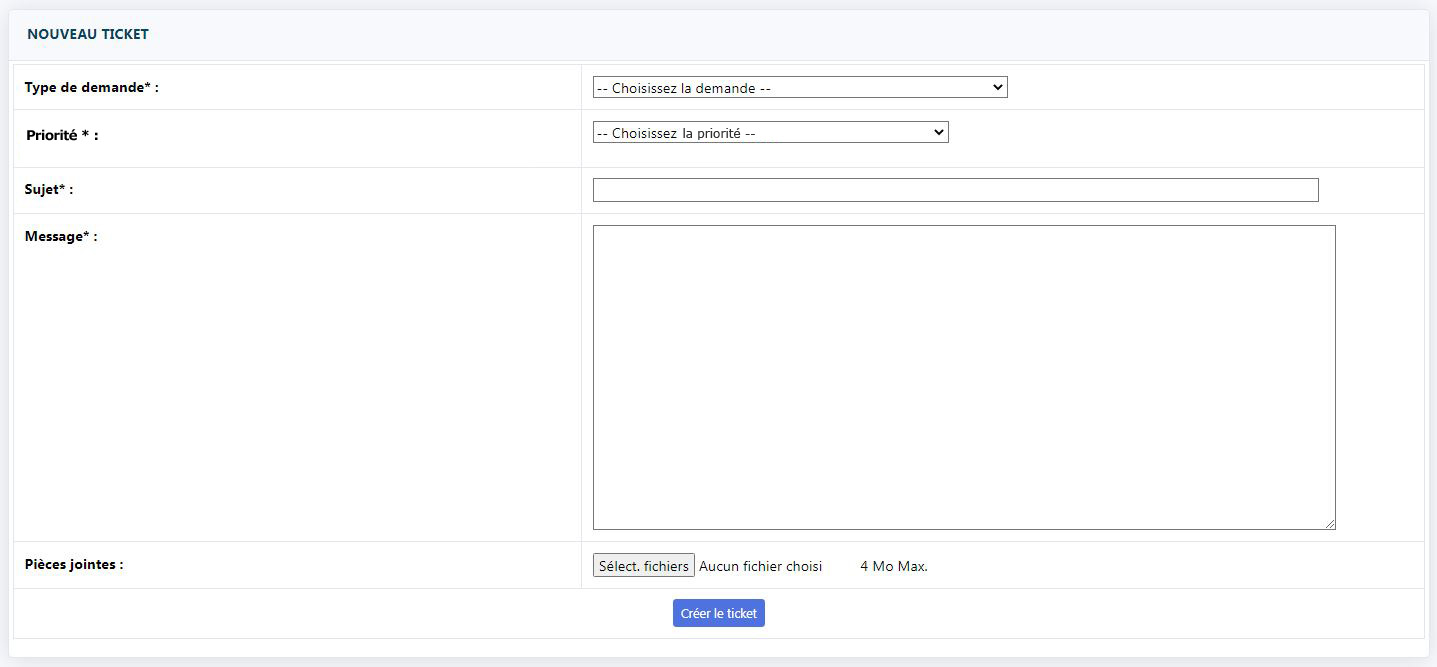
Des annexes pourront être rajoutées au présent document en fonction des remarques ou évolutions proposées au cours de l’année.

ANNEXES

Capture 1 : page listing des tickets



Capture 2 : ouverture d’un nouveau ticket



Capture 3 : cycle de vie d’un ticket

