

At-Bénévoles

1. Contexte

L'association Stras'Orgues organise chaque année depuis 5 ans, fin août un festival de musiques d'orgue à Strasbourg. Plus d'une trentaine de bénévoles font vivre le festival durant une semaine : transport de matériel, mise en place des chaises et bancs, ventes et contrôle de billets, sécurité... Le festival dure une semaine, mais les bénévoles sont mobilisés sur une dizaine de jours.

2. Mission

La gestion des bénévoles est actuellement réalisée dans un tableau Excel qui atteint ses limites. Le but du projet est de réaliser cette gestion dans une application à interface web. Les données seront conservées dans une base MariaDB et Laravel sera utilisé pour la construction de l'application. L'application doit permettre :

- de renseigner la période d'activité des bénévoles pour le festival,
- de saisir / modifier les données des différents bénévoles : prénom, nom, téléphone et adresse électronique, compétences (types de poste possible),
- de saisir les créneaux de disponibilité de chaque bénévole par demi-heure sur la période du festival,
- de saisir les lieux, les concerts, les types de poste,
- de saisir les différents postes de travail : nom du poste, date, créneau horaire, type de poste, lieu, concert,
- d'affecter un ou plusieurs bénévoles à un poste avec une recherche automatique de bénévole tenant compte de ses disponibilités restantes (l'affectation à un poste réduit les disponibilités), compétences,
- de visualiser l'enchaînement des tâches d'un bénévole,
- d'éditer des listes : liste de bénévoles, tâches par bénévole, bénévoles par type de tâche, bénévoles par jour, tâches par jour...

L'accès à l'application se fait en s'identifiant sur le principe de comptes individuels (l'identifiant sera une adresse de courrier électronique). Deux types de comptes sont à prévoir : gestionnaire et bénévole. Le gestionnaire a accès à toutes les données et fonctionnalités. Chaque bénévole pourra avoir son compte pour renseigner ses disponibilités et visualiser ses tâches.

3. Compléments

La méthode de travail tend à se placer dans des conditions professionnelles. Un certain nombre de fondamentaux sont donc à respecter : démarche de gestion de projet, règles de développement, attention particulière à la documentation.

3.1. Gestion de projet

Une démarche de gestion de projets implique la définition de tâches, leur attribution à une personne, la définition d'échéances, un plan de charge, des états d'avancement, une gestion du temps, un travail d'équipe... GitHub propose un outil de gestion de projet : les tickets et jalons devront donc être utilisés pour le projet, et renseignés au fur et à mesure de son avancement. Le travail de planification doit évidemment débuter lors de la première séance. De plus, chaque démarche d'autoformation individuelle devra être notée en tant que tâche dans GitHub.

Le but des ateliers de professionnalisation est d'aboutir sur les épreuves du BTS, dans les meilleures conditions possible. L'évaluation individuelle des projets sera réalisée selon plusieurs critères :

- l'implication dans le travail d'équipe,
- le partage de connaissances,
- l'organisation personnelle,
- l'effort d'autoformation,
- le respect des délais,
- la qualité du résultat obtenu.

3.2. Règles de développement

Dans le cadre d'une approche d'équipe, un développement raisonnable ne peut se concevoir sans règles de développement. Pour les ateliers de professionnalisation :

- le code source doit être versionné dans GitHub,
- les règles de codage à respecter sont celles du PHP-FIG (Framework Interoperability Group), <https://www.php-fig.org>

3.3. Documentation

La documentation constitue un élément vital d'un projet, trop souvent négligé. Un formalisme doit être défini, avec un modèle de document comportant des données de base : auteur, date de rédaction, date de mise à jour, titre, pagination. Comme pour le code, une gestion des versions est indispensable, car les modifications de la documentation sont à prévoir nécessairement en parallèle avec celles des fonctionnalités et donc du code.

La documentation se subdivise en deux catégories : la documentation technique et la documentation d'utilisation, elle-même subdivisée en fonction du niveau : administrateur, utilisateur... Chacune de ces catégories se destine à un autre public. Une réflexion doit être menée pour la rendre accessible de manière appropriée pour chaque destinataire. Ce moyen n'est pas nécessairement le même pour chacun d'eux.

3.3.1. Documentation technique

La documentation technique est destinée aux développeurs et mainteneurs de l'application. Dans le cadre des Ateliers de professionnalisation, les rédacteurs des cahiers des charges ont aussi accès au développement. Le but de la documentation technique est

de permettre de comprendre le code, de faciliter les corrections, les évolutions... Pour les Ateliers de professionnalisation, les documents seront rédigés avec le formalisme d'UML2. La documentation technique comportera au minimum les documents suivants :

- le résumé des normes de codage,
- les prérequis d'installation (logiciels et technologies utilisées),
- la procédure d'installation et de mise à jour,
- le diagramme de contexte statique,
- les diagrammes de cas d'utilisation,
- les descriptions textuelles des cas d'utilisation,
- le diagrammes de classes,
- le modèle entité-associations pour les bases de données.

Selon leur utilité pour la compréhension du projet, les diagrammes suivants pourront aussi être intégrés :

- un ou plusieurs diagrammes de séquence système,
- un ou plusieurs diagrammes d'activité,
- un ou plusieurs diagrammes de packages.

3.3.2. Documentation utilisateur

Les utilisateurs de l'application sont à regrouper en fonction de leur responsabilité : administrateur, administrateur réseau et utilisateur final. La documentation utilisateur doit toujours s'adapter à son lecteur et être facilement accessible. Ainsi les administrateurs n'auront pas les mêmes besoins que les utilisateurs finaux. Une réflexion pourra donc être menée avec un choix entre une documentation en ligne ou un document au format PDF. Dans tous les cas, la documentation utilisateur devra comporter :

- des consignes de configuration, d'installation ou de mise à jour,
- un manuel d'administration,
- un manuel d'utilisation de l'outil de suivi de bugs et de demandes d'évolutions,
- un manuel utilisateur.