

Offre de stage sur un logiciel Open Source — GVV (BUT2)

Date : 2026-02-22

Public visé : Étudiantes/étudiants de 2e année de BUT

Cadre : Contribution Open Source, organisation Agile Scrum, avec **intégration continue**.

1) Présentation de GVV

GVV (Gestion Vol à voile) est une application Open Source et gratuite destinée à **la gestion opérationnelle et administrative des aéroclubs** depuis 2011. Elle permet de gérer simultanément plusieurs activités (planeur, avion, ULM) tout en maintenant une séparation complète des comptes de chaque type d'activité.

Elle couvre notamment :

- **Gestion des membres et des inscriptions** : suivi des adhésions, autorisations parentales pour mineurs, historique des activités.
- **Flotte d'aéronefs et maintenance** : suivi des appareils, calendrier de maintenance, heures de vol, entretiens.
- **Gestion des vols** : enregistrement des vols effectués, facturation automatique, statuts des parcours, calendrier partagé.
- **Système de facturation et comptabilité** : génération de factures, suivi des paiements, rapprochement bancaire, écritures comptables simples.
- **Communications** : envoi d'emails groupés, gestion des listes de diffusion, intégration avec OpenFlyers.

GVV est utilisée par 5–6 associations de vol à voile en France et Belgique. Déployée depuis 15 ans en production, elle est toujours en développement actif, avec une attention particulière sur la **stabilité et la qualité du code**.

Le stage consiste à contribuer à des fonctionnalités réellement utiles au projet, avec le cycle de développement d'une équipe logicielle professionnelle.

2) Workflow de développement : CI avec Jenkins et branche “main” toujours fonctionnelle

GVV est développé avec une contrainte forte : **la branche main doit rester continuellement fonctionnelle.**

Conséquence directe pour le stage :

- privilégier des PRs **petites et fréquentes**, faciles à relire et à valider ;
 - écrire/mettre à jour des tests;
 - documenter les changements
 - Le stage sensibilise donc à une logique “CI/CD” :
-

3) Organisation : 4 sprints de 2 semaines en Agile (Scrum)

Le stage est organisé en **4 sprints de 2 semaines** (8 semaines), selon Scrum.

Valeurs et principes

- **Transparence** : tickets, PRs, résultats Jenkins et avancement visibles.
- **Inspection** : revues de code + tests + démonstrations en fin de sprint.
- **Adaptation** : on ajuste le plan selon les retours et les contraintes techniques.

Rôles et rituels

- **Product Owner (PO)** : priorise et clarifie les besoins (liés aux Epics).
- **Scrum Master / référent** : aide à lever les blocages, veille au cadre.
- **Équipe de dev** : réalise les items, garantit la qualité sur **main**.

Rituels adaptés au contexte stage :

- **Sprint Planning** : objectif de sprint + sélection d'issues.
- **Point court journalier** : ce que j'ai fait hier / ce que je vais faire aujourd'hui / blocages.
- **Sprint Review** : démonstration de ce qui est terminé (sur l'environnement de test).
- **Rétrospective** : amélioration continue (qualité, CI, organisation, estimation).

Definition of Done (exemple) Un item est “terminé” quand :

- PR revue et approuvée
 - code mergé sur **main**,
 - (tests OK),
 - fonctionnalité démontrable,
 - documentation si nécessaire.
-

4) Trois missions distinctes

Chaque mission sera découpée en tickets progressifs.

Mission A --- Gestion documentaire

Contexte : L'administration nous impose de plus en plus de contraintes réglementaires. Le suivi des documents et de leur date d'expiration est complexe et fastidieux. Le but est de donner une vue simple aux administrateurs sur les prochaines échéances et de garantir la conformité.

Feature --- Archivage documentaire (existant)

- Mettre en place l'archivage compressé des documents
- Permettre la visualisation en ligne des documents archivés
- Permettre le téléchargement des documents archivés

Feature --- Approbation et signature

- Permettre la demande d'approbation d'un document
- Permettre la signature par un membre
- Permettre la signature par un intervenant externe
- Suivre l'état d'approbation / signature

Feature --- Génération documentaire

- Générer des documents PDF à partir des données applicatives
- Concevoir un moteur de templates configurables
- Permettre la configuration des documents sans développement
- Interface d'administration pour définir les modèles

Feature --- Gestion des procédures

- Définir une structure de workflow configurable
- Associer pages et documents aux étapes
- Gérer les documents à consulter / fournir / approuver
- Interface d'édition de procédures sans code
- Exécuter et suivre une procédure

Mission B --- Facturation et paiements en ligne

Contexte :

Actuellement, GVV gère des comptes pilotes et les adhérents doivent régler par chèque ou virement. L'intégration d'un paiement en ligne (ex. Stripe, PayBox, PayPal) réduirait les délais et simplifierait la comptabilité.

Feature --- Paiement en ligne

- Intégrer PayPal
- Intégrer paiement carte bancaire
- Intégrer HelloAsso ou prestataires équivalents
- Gérer la confirmation et la traçabilité des paiements

Feature --- Vols de découverte

- Permettre l'achat en ligne
- Générer automatiquement les bons
- Gérer leur validation / utilisation

Feature --- Configuration de facturation

- Permettre le paramétrage des règles de facturation
- Exprimer les règles de facturation sans code PHP (pour l'instant c'est codé en dur)
- Interface d'administration

Feature --- Adaptation clubs

- Paramétriser les bons de vols
- Supporter les variantes de fonctionnement
- Gestion multi-configuration

Mission C -- Gestion de la maintenance

Feature -- Gestion de la flotte

- Fiche d'information relative à la maintenance
- liste des équipements
- règles des butées calendaires ou par heures de vol

Feature gestion des programmes de maintenance

- Création, modification, versioning des programmes
- Suivi des bulletins de service

Feature planification de la maintenance

- Calendrier des visites
- Suivi des heures restantes avant maintenance

Feature enregistrement des opérations de maintenance

- Enregistrement des opérations de maintenance
- Fiches d'interventions

Feature Audit / Rapports

- Rapport des opérations de maintenance

6) Profil recherché

- BUT2 : bases en programmation, web/logiciel
- rigueur et envie d'apprendre (tests, CI, code review)
- capacité à communiquer et à demander de l'aide tôt
- intérêt pour l'Open Source et le travail en méthode Agile

7) Ce que le stage apporte

- expérience concrète en workflow GitHub (issues, PR, review)
- pratique de Scrum sur 4 sprints
- compréhension d'un projet où **main** doit rester stable
- exposition à la CI (Jenkins) et à une logique de release "sur demande"