

Rapport de la semaine 26 mai - 01 juin

Application Shiny du Rotaract Club de Delmas

Date : 03/06/2025

Période Couverte : Depuis le 26 mai 2015 jusqu'à aujourd'hui

Préparé par : Guy Junior CALVET

À l'attention de : Aurélien NICOSIA

Objet : Rapport hebdomadaire sur l'état d'avancement et des fonctionnalités de l'application Shiny pour la gestion de la base de données du Rotaract Club de Delmas.

Introduction

Cette semaine a été marquée par une phase significative de reconception de l'architecture de notre application Shiny. L'objectif principal était d'intégrer un système d'authentification robuste et une gestion dynamique des rôles utilisateurs, en lien direct avec la base de données MySQL du Club Rotaract de Delmas. Cette refonte a été initiée pour répondre aux exigences de sécurité et de personnalisation de l'accès aux différentes fonctionnalités de l'application.

1. Problèmes Rencontrés Durant la Semaine

1.1. Intégration de l'authentification avec shinymanager

Le défi majeur de la semaine a été l'implémentation de l'authentification des utilisateurs via le package shinymanager.

- **Conflits de version Bootstrap :** Une difficulté initiale est survenue lors de l'intégration de shinymanager avec la structure de l'interface utilisateur basée sur bslib et Bootstrap 5. La page de connexion de shinymanager, si configurée avec des thèmes shinythemes (basés sur des versions antérieures de Bootstrap), entrainait en conflit avec le thème Bootstrap 5 du reste de l'application, générant des erreurs de rendu et des avertissements critiques (session\$setCurrentTheme() cannot be used to change the Bootstrap version).
- **Configuration de check_credentials_db :** La configuration de shinymanager pour qu'il authentifie les utilisateurs directement à partir de la base de données MySQL existante (table Utilisateurs ou Membres) plutôt qu'à partir d'un data.frame en mémoire ou d'un fichier SQLite géré par shinymanager lui-même, a nécessité plusieurs itérations. Des erreurs de type "Operation not allowed without an active reactive context" sont apparues, liées à la manière dont la connexion à la base de données était établie et passée à shinymanager.
- **Récupération et gestion dynamique des rôles :** Assurer que le rôle de l'utilisateur (ex: Président, Secrétaire), une fois l'authentification réussie via

la base de données, soit correctement récupéré et utilisé pour adapter dynamiquement l'interface utilisateur (menus, accès aux pages) a été un point d'attention.

1.2. Reconception de l'Interface Utilisateur (UI)

L'introduction de l'authentification et des rôles a nécessité de repenser la structure globale de l'UI :

- Passage d'une structure potentiellement plus simple à une architecture multi-pages dynamique (page_navbar avec uiOutput pour les menus conditionnels).
- Assurer la cohérence du design (thème jour/nuit, couleurs dominantes rouge et blanc) à travers toutes les pages, y compris la page de connexion et les pages post-connexion.

2. Méthodes Développées pour Résoudre les Problèmes

2.1. Gestion des conflits de thèmes Bootstrap

- La solution a consisté à s'assurer que l'ensemble de l'application (y compris l'UI principale après connexion) est initialisée avec un thème bslib Bootstrap 5.
- Le thème shinythemes a été retiré de la configuration de la page de connexion de secure_app() pour éviter les conflits de version.
- L'appel initial à session\$setCurrentTheme() dans le serveur pour le thème par défaut a été supprimé, le thème étant désormais défini directement dans la page_navbar de l'UI.

2.2. Configuration de check_credentials_db

- Après plusieurs essais, l'approche consistant à passer les arguments de connexion (conn_args) directement à check_credentials_db plutôt qu'un objet de connexion pré-établi s'est avérée plus stable pour l'initialisation de shinymanager. Cela permet à shinymanager de gérer lui-même le cycle de vie de cette connexion spécifique à l'authentification.
- Une attention particulière a été portée à la configuration des db_infos (noms de table, colonnes utilisateur, mot de passe, et colonnes additionnelles comme Role).

2.3. Refonte de l'UI et gestion des rôles

- Utilisation de uiOutput dans le serveur pour générer dynamiquement les nav_panel du menu en fonction du rôle récupéré après authentification.
- Définition claire des thèmes light_theme et dark_theme avec bslib pour assurer la cohérence visuelle.

3. Apprentissages Effectués

- Compréhension approfondie du fonctionnement interne de shinymanager, notamment les différentes méthodes de check_credentials et leurs implications (en mémoire, SQLite géré, base de données externe).
- Maîtrise accrue de bslib pour la gestion des thèmes Bootstrap 5 dans Shiny, y compris la gestion des versions et l'application dynamique de thèmes.
- Techniques de débogage des contextes réactifs dans Shiny, particulièrement lors de l'intégration de modules ou de fonctionnalités qui interagissent tôt dans le cycle de vie de la session.
- Importance de la structuration du code pour séparer les préoccupations (connexion DB pour authentification vs connexion DB pour les opérations de l'application).
- Prise de conscience de la complexité ajoutée par la gestion des identités et des accès dans une application web, et de l'importance de planifier cette intégration dès les premières phases de conception.

4. Réalisations de la Semaine

- **Mise en place d'une structure d'authentification fonctionnelle avec shinymanager :** Bien qu'encore en cours de finalisation pour la connexion à la base de données MySQL, la logique de base et la gestion des erreurs de configuration ont progressé.
- **Refonte majeure de l'interface utilisateur :**
 - Création d'une page d'accueil conforme aux spécifications (logo, titre, informations développeur).
 - Implémentation d'un menu hamburger dynamique basé sur les rôles utilisateurs (logique serveur prête, en attente de la récupération effective des rôles depuis la DB).
 - Mise en place du mécanisme de basculement entre thème jour et thème nuit sur toutes les pages.
- **Intégration initiale des pages existantes dans la nouvelle structure :** Les codes pour "Vues d'informations", "Saisie de données" et "Requêtes SQL avancées" ont été adaptés pour s'insérer comme des nav_panel dans la nouvelle architecture.
- Le code de l'application a été restructuré pour une meilleure lisibilité et maintenabilité, notamment en ce qui concerne la gestion des thèmes et la logique d'authentification.

5. Objectifs pour la Semaine Prochaine

- Finaliser et stabiliser l'authentification des utilisateurs via shinymanager en se connectant directement à la table Utilisateurs (ou équivalente) de la base de données MySQL.
 - S'assurer de la récupération correcte du rôle de l'utilisateur.
 - Tester rigoureusement les différents scénarios de connexion (utilisateur valide, invalide, différents rôles).
- Rendre le menu déroulant et l'accès aux pages pleinement fonctionnels en fonction du rôle de l'utilisateur authentifié.
- Poursuivre l'intégration et l'adaptation fine des fonctionnalités existantes dans les pages "Vues d'informations", "Saisie de données", et "Requêtes".
- Commencer le développement de la page "Requêtes" (prédéfinies).

Conclusion

La reconception de l'interface et l'intégration de l'authentification ont été plus chronophages que prévu initialement, en raison de la complexité de faire interagir shinymanager avec une base de données externe et les spécificités de bslib. Cependant, cette base solide est essentielle pour la sécurité et la convivialité de l'application. Le code existant pour les fonctionnalités métier reste pertinent et son intégration dans la nouvelle structure est en bonne voie.

