

# Projet Festival

## Cahier des charges et méthodologie

### Table des matières

I - Contexte.....	3
II - Cahier des charges.....	3
III - Gestion de projet.....	7
IV - Les tâches du projet par itération.....	13
V - Annexes.....	29

## I - Contexte

La ville de Saint Malo organise chaque année un festival de musique intitulé « Folklores du monde » et s'appuie sur une application web nommée « Festival » pour la gestion des hébergements des groupes participant.

Vous êtes chargés de faire évoluer cette application dont le contexte est décrit dans le document «2SLAM\_PPE\_C1\_festival\_contexte ».

- Maître d'ouvrage : la mairie de Saint Malo.
- Maître d'œuvre : les enseignants de PPE.

## II - Cahier des charges

### II.1 Définition du besoin

L'application actuelle doit évoluer selon deux axes principaux :

- enrichissement fonctionnel :
  - gestion des groupes musicaux
  - gestion de la programmation du festival.
- sécurisation de l'accès à l'application web intranet : l'étape d'authentification sera perfectionnée (rôles)..

### II.2 Ressources

Un dossier initial vous est fourni, comprenant :

- Les éléments de description du contexte,
- Les fichiers nécessaires à l'installation de l'application actuelle sous la forme d'un projet Netbeans :
  - le projet initial est à récupérer sur un dépôt à l'adresse suivante : <https://framagit.org/nbourgeois/festivalphp2020.git>
  - En plus du code source, le projet contient :
    - une base de données de test : script SQL pour MySQL dans le répertoire *sql* du projet
    - une documentation générée (générée par APIGen) de l'API de la couche modele, dans le répertoire *doc* du projet
- Les diagrammes initiaux, de cas d'utilisation, de classes , de données et de Gantt
- Une documentation technique (fichier *2SLAM\_PPE\_P1\_Doc\_Tech\_Festival\_intranet.pdf*) permet d'expliquer les choix d'architecture logicielle pour :
  - comprendre le fonctionnement de l'application
  - les respecter lors des opérations de maintenance

### II.3 Objectifs

#### II.3.1 - Objectifs de déploiement

A terme, l'application sera déployée sur le serveur web de production du BTS SIO. Ce serveur est commun à toute la section et donc accessible au maître d'œuvre. Il sera aussi le serveur de test pour l'examen E4 du BTS.

#### II.3.2 - Objectifs de qualité du code

- Les noms de fichiers, de classes, d'objets, de méthodes, de variables et de constantes doivent respecter les normes étudiées en cours.
- Vous utiliserez l'environnement de développement NetBeans.
- Vous commenterez votre code avec les annotations (« tags ») PhpDoc.
- La couche métier doit mettre en œuvre la programmation orientée objet.
- Le code ajouté doit respecter l'architecture du code existant

### ***II.3.3 - Objectifs de sécurité***

Les valeurs saisies dans les formulaires doivent être vérifiées, **avant** insertion dans la base de données , en terme de :

- vraisemblance
- respect des format et des domaines de valeur
- protection contre les attaques par injection SQL

### ***II.3.4 - Objectifs d'ergonomie***

L'ergonomie générale de l'application actuelle sera conservée. Des améliorations, à soumettre au maître d'œuvre, sont possibles, mais cela ne constitue pas un objectif prioritaire.

Par ailleurs, on souhaite que l'utilisateur soit averti dès qu'il quitte un champ de saisie si la valeur est erronée (javascript). Cela ne doit pas remplacer la vérification du formulaire en PHP lors de sa validation.

## III - Gestion de projet

### III.1 Méthodes de travail

Les projets doivent être conduits en équipes, suivant une méthode itérative, empruntant quelques éléments de méthodes agiles. L'équipe de développement s'auto-organise.

- Une itération doit toujours conduire à un produit livrable et documenté, destiné au maître d'ouvrage.
- Une itération peut durer une ou plusieurs séances.
- A chaque séance, l'équipe doit planifier son travail lors d'une courte réunion (« daily scrum »). **La trace de ces « daily - scrum » doit être conservée.**
- Le daily-scrum doit notamment contenir le bilan de la séance précédente, les décisions de l'équipe comme la répartition des tâches pour la séance en cours.
- L'équipe désigne un chef d'équipe (le « scrum master »).
- L'outil de gestion de projet est «**Trello**».
  - Pendant tout le projet, chaque collaborateur est responsable de ses tâches trello et de leur avancement.
  - La maîtrise d'œuvre doit être en mesure, à chaque instant, de savoir :
  - Pour une tâche :
    - qui est affecté
    - la date d'échéance
    - le statut (à faire, en cours, terminée)
    - le degré d'avancement (utiliser la checkList)
  - Le daily-scrum est, entre autres l'occasion de vérifier l'affectation des tâches et les mettre à jour éventuellement.
  - En fin de séance, les collaborateurs doivent mettre à jour leur tâches Trello.
- Le scrum-master doit notamment :
  - animer le travail d'équipe et notamment les daily-scrums
  - constituer son équipe sur l'outil de gestion de projet ( redmine )
  - affecter les tâches aux développeurs sur l'outil de gestion de projet
  - rendre compte au maître d'œuvre en cas de besoin
- Les développeurs doivent :
  - renseigner l'avancement de leurs tâches sur l'outil de gestion de projet
  - respecter les conventions de travail collaboratif (usage des outils de gestion de projet, de gestion des versions, ...)
- Itérations :
  - La première itération conduit généralement à la production d'un diagramme de Gantt prévisionnel pour l'ensemble du projet. Veillez à l'enregistrer dans sa version initiale, Le diagramme de Gantt devra être réactualisé après chaque séance (diagramme de Gantt réel).
  - En début d'itération, les tâches et sous-tâches doivent être réparties et le planning prévisionnel adapté.
  - En fin d'itération, un dossier documentaire doit être déposé sur Moodle et une branche identifiable du dépôt de code doit matérialiser la version du projet correspondant à cette itération.
- Le maître d'ouvrage dépend du contexte étudié.
- Les maîtres d'œuvre sont les enseignants de PPE.

### **III.2 Définition d'un « livrable »**

A chaque itération, le maître d'œuvre indique à l'équipe de développement la liste des productions qui constitueront le livrable. Le scrum-master chargera un membre de l'équipe de veiller à la constitution du livrable à l'échéance prévue.

Le contenu d'un « livrable » contiendra des productions parmi les suivantes :

- l'application déployée et utilisable sur un serveur d'application faisant office de plate-forme de test.
  - dans le cas d'une application Web : l'URL de l'application sur une plate-forme de test fonctionnel
  - dans le cas d'un client lourd ou mobile : une machine virtuelle sur laquelle l'application est installée
- un mode opératoire de démarrage et d'utilisation
- l'application est accompagnée des informations nécessaires pour la tester : jeu d'essai, avec les codes d'accès des différents types d'utilisateurs ;
- le code source de l'application
  - sur un dépôt de code accessible au maître d'œuvre ;
  - le code est disposé et commenté suivant des normes décidées en accord avec le maître d'œuvre
- les documents de spécification :
  - un diagramme des cas d'utilisation
  - les scénarios des fonctionnalités étudiées
  - le jeu d'essai prévu pour tester les scénarios
  - les maquettes d'interfaces graphiques des fonctionnalités étudiées
  - le(s) schéma(s) de la bases de données de l'application : modèle relationnel, MCD, ...
  - le diagramme de classes
  - autres : en fonction de la demande
- les documents relatifs à la gestion de projet
  - le diagramme de Gantt prévisionnel
  - le diagramme de Gantt actualisé
  - la répartition des tâches et des fonctions
  - la trace des scrums
- les rapports de tests
- les besoins en formation et en auto-formation
- autres

Les documents constituant le livrable seront au format PDF.

### **III.3 Les tests**

#### **III.3.1 - Tests unitaires**

les tests unitaires des classes métier et modèle seront systématiques (une classe de test par classe codée). Il faut concevoir les classes de test au fur et à mesure.

#### **III.3.2 - Tests fonctionnels**

les tests fonctionnels seront réalisés en utilisant les scénarios et les jeux d'essai prédéfinis ;  
les tests donneront lieu à des rapports de tests.

#### **Rappels :**

2. Les **scénarios** et les **jeux d'essai** sont conçus **avant** la réalisation de l'application (codage et tests)
3. 1 **cas d'utilisation** est décrit par plusieurs scénarios : 1 scénario nominal, des scénarios alternatifs, des scénarios d'exception
4. 1 jeu d'essai est destiné à valider l'ensemble des scénarios **d'un** cas d'utilisation
5. 1 cas de test est un ensemble de valeurs destiné à tester un scénario
6. 1 scénario peut être validé par plusieurs cas de test
7. L'ordre des cas de test est important, car un cas de test peut affecter l'état initial de la base de données
8. 1 jeu d'essai fait l'objet d'un seul test par **rapport de tests** : ce test est soit conforme, soit non conforme (dans ce cas, un commentaire est utile)
9. Le rapport de tests comporte une ligne de tableau par test ;  
se reporter aux modèles de documents fournis.

## IV - Les tâches du projet par itération

Pour faciliter le travail des équipes, les maîtres d'œuvre ont pré-établi les tâches et leur répartition pour chaque itération du projet. Cette présentation sera précisée par un diagramme de Gantt.

### *IV.1 Liste des itérations*

La version de l'application qui vous est fournie possède deux caractéristiques essentielles : elle est "objet" et suit le modèle MVC. On vous demande de faire quelques modifications graphiques, de résoudre quelques dysfonctionnements et d'ajouter un certain nombre de fonctionnalités.

- Itération 1 : Installation, test fonctionnel de l'existant, modifications graphiques, résolution de pannes
- Itération 2 : Consultation des groupes
- Itération 3 : Mise à jour (insertion/modification/suppression) d'un groupe
- Itération 4 : Modification de la modélisation objet et modélisation BDD en prenant en compte les représentations.
- Itération 5 : Nouvelles classes métiers et DAO, et consultation des représentations
- Itération 6 au choix :
  - Insertion/modification/suppression d'une représentation
  - Gestion des rôles utilisateurs

### IV.2 Itération n°1 - Installation, tests, maintenance

Dans un premier temps, on vous demande d'organiser votre équipe de projet et votre environnement de développement, de faire quelques modifications graphiques, de résoudre quelques dysfonctionnements.

Cela vous permettra de vous familiariser avec l'architecture logicielle.

	Tâche	E1	E2	E3	Production	Activité
	<b>Itération 1</b>					
T1	Analyse de la demande et de l'existant					
T1_1	Analyser le cahier des charges	X	X	X		A1.1.1 Analyse du cahier des charges d'un service à produire
T1_2	Créer un dépôt sur Framagit et inscrire les collaborateurs et les professeurs	X				A1.4.1 Participation à un projet A5.1.1 Mise en place d'une gestion de configuration
T1_3	Créer un projet trello et les cartes, et inviter les collaborateurs et les professeurs		X			A1.4.1 Participation à un projet
T1_4	Récupérer l'application	X	X	X		A4.1.6 Gestion d'environnements de développement et de test
T1_5	Installer la base de données en local	X	X	X		A4.1.3 Conception ou adaptation d'une base de données
T1_6	Tester les classes métier	X	X	X	Rapport de test unitaire (se répartir les classes à tester)	A4.1.8 Réalisation des tests nécessaires à la validation d'éléments adaptés ou développés
T1_7	Tester les classes DAO	X	X	X	Rapport de test unitaire(se répartir les classes à tester)	A4.1.8 Réalisation des tests nécessaires à la validation d'éléments adaptés ou développés
T1_8	Tester l'application en local	X	X	X	Rapport de test fonctionnel	A1.3.1 Test d'intégration et d'acceptation d'un service
<b>TI_9</b>	<b>Documenter l'application réaliser un diagramme de séquence (en s'aidant du débogueur) pour les 2 scénarios nominaux suivants :</b> <ul style="list-style-type: none"> <li>• consulter un établissement</li> <li>• modifier un établissement</li> </ul>		X	X	<b>2 diagrammes de séquence</b>	
T2	Résolution d'incidents					A2.2.1 Suivi et résolution d'incidents
T2_1	Résoudre fiche incident 24	X			Fiche résolution incident	A4.2.1 Analyse et correction d'un dysfonctionnement, d'un problème de qualité de service ou de sécurité
T2_2	Résoudre fiche incident 25		X		Fiche résolution incident	
T2_3	Résoudre fiche incident 26			X	Fiche résolution incident	
T2_4	Résoudre fiche incident 27			X	Fiche résolution incident	
T2_5	Résoudre fiche incident 28		X		Fiche résolution incident	
T2_6	Résoudre fiche incident 29	X			Fiche résolution incident	
T3	Modification charte graphique					
T3_1	Personnaliser l'interface (couleur de fond, logo)	X			Copies d'écran	
	<b>Évaluation 1 par les professeurs</b>				<b>Note 1</b>	



**IV.2.1 - Itération n°2 - consultation des groupes**

On souhaite l'ajout d'une fonctionnalité permettant de consulter la liste des groupes enregistrés dans la bases de données.

	Tâche	E1	E2	E3	Production	Activité
	<b>Itération 2</b>					
T4	Ajout de la fonctionnalité : consultation des groupes					
T4_1	Modifier le diagramme de cas d'utilisation		X		Document comprenant le diagramme	
T4_2	Création de la vue des groupes	X		X	Code sur dépôt git	
T4_3	Création du contrôleur		X		Code sur dépôt git	
T4_4	Intégration de la fonctionnalité à l'application		X		Code sur dépôt git	
T4_5	Tester l'intégration de la fonctionnalité	X		X	Rapport de test fonctionnel	

### IV.2.2 - Itération n°3 - mise à jour des groupes

On souhaite compléter la fonctionnalité de gestion des groupes en permettant la mise à jour (insertion/modification/suppression) d'un groupe.

	Tâche	E1	E2	E3	Production	Activité
	<b>Itération 3</b>					
<b>T5</b>	Ajout de la fonctionnalité : mise à jour des groupes					
T5_1	Modifier le diagramme de cas d'utilisation		X		Document comprenant le diagramme	
T5_2	Création de la vue montrant le formulaire d'ajout/modification d'un groupe	X		X	Code sur dépôt git	
T5_3	Compléter le contrôleur (ajout/modification)	X		X	Code sur dépôt git	
T5_4	Intégration de la fonctionnalité (ajout/modification) à l'application	X		X	Code sur dépôt git	
T5_5	Tester l'intégration de la fonctionnalité		X		Rapport de test fonctionnel	
T5_6	Création de la vue de suppression d'un groupe		X		Code sur dépôt git	
T5_7	Compléter le contrôleur		X		Code sur dépôt git	
T5_8	Intégration de la fonctionnalité à l'application		X		Code sur dépôt git	
T5_9	Tester l'intégration de la fonctionnalité	X		X	Rapport de test fonctionnel	
	<b>Évaluation 2 par les professeurs</b>				<b>Note 2</b>	

### IV.2.3 - Itération n° 4 - gestion des représentations - évolution du modèle

On souhaite que la programmation du festival soit accessible aux visiteurs du site. On considère qu'un spectacle donné dans le cadre du festival est caractérisé par un groupe, un lieu, une date, une heure de début, une heure de fin. Chaque lieu a un nom, une adresse, une capacité d'accueil.

Quelques contraintes :

pas de recouvrement de 2 représentations dans un même lieu.

pas de recouvrement de 2 représentations d'un même groupe.

l'heure de fin d'un spectacle doit avoir lieu avant 24:00

**Travail à faire :**

compléter le diagramme de classes ainsi que le MLDR,

ajouter et modifier les tables nécessaires,

créer et modifier les classes nécessaires,

créer et exécuter les tests unitaires, rédiger les rapports de tests unitaires

	Tâche	E1	E2	E3	Production	Activité
	<b>Itération 4</b>	consultation des représentations : adaptation du modèle				
<b>T6</b>	Modélisation des représentations et des lieux					
T6_1	Adapter le diagramme de classes métier	X			Document comprenant le nouveau diagramme de classes	
T6_2	Adapter le MLDR			X		
<b>T7</b>	Les lieux					
T7_1	Modifier la structure de la base de données			X	script SQL de création de la BDD sur dépôt Git	
T7_2	Peupler la nouvelle table			X	script SQL de création de la BDD sur dépôt Git	
T7_3	Coder la classe Lieu	X			Code sur dépôt git	
T7_4	Coder classe TestLieu et l'exécuter		X		Code sur dépôt git Rapport de test	
T7_5	Coder classe DaoLieu			X	Code sur dépôt git	
T7_6	Coder classe TestDaoLieu et l'exécuter		X		Code sur dépôt git Rapport de test	
<b>T8</b>	Les représentations					
T8_1	Modifier la structure de la base de données	X			script SQL de création de la BDD sur dépôt Git	
T8_2	Peupler le(s) table(s)	X			script SQL de création de la BDD sur dépôt Git	
T8_3	Coder classe Representation			X	Code sur dépôt git	
T8_4	Coder classe TestRepresentation et l'exécuter			X	Code sur dépôt git Rapport de test	
T8_5	Coder classe DaoRepresentation	X	X		Code sur dépôt git	
T8_6	Coder classe TestDaoRepresentation et l'exécuter	X	X		Code sur dépôt git Rapport de test	

**IV.2.4 - Itération n°5 - gestion des représentations - consultation**

Il s'agit maintenant de programmer la fonctionnalité de consultation des représentations et de l'intégrer à l'application : ajouter une page de consultation des représentations, groupées par lieu et par date.

	Tâche	E 1	E 2	E 3	Production	Activité
	<b>Itération 5</b>	Ajout de la fonctionnalité : consultation des représentations				
<b>T9</b>						
T9_1	Création de la vue de consultation des représentations		X	X	Code sur dépôt git	
T9_2	Création du contrôleur correspondant	X		X	Code sur dépôt git	
T9_3	Intégration de la fonctionnalité à l'application	X		X	Code sur dépôt git	
T9_4	Tester l'intégration de la fonctionnalité		X		Rapport de test fonctionnel	
<b>T10</b>	<b>Déploiement</b>					
T10_1	Déployer l'application sur le serveur de test de l'équipe	X	X			
T10_2	Tester l'application déployée			X	Rapport de test	

### IV.2.5 - Itération n°6 - ajout d'une fonctionnalité au choix

#### IV.2.5.1 Mise à jour des représentations

On souhaite maintenant pouvoir ajouter, modifier et supprimer des représentations.

	Tâche	E1	E2	E3	Production	Activité
	<b>Itération 6</b>					
<b>T11</b>	Ajout de la fonctionnalité : mise à jour des représentations					
T11_1	Modifier le diagramme de cas d'utilisation		X		Document comprenant le diagramme	
T11_2	Création de la vue montrant le formulaire d'ajout/modification d'une représentation	X		X	Code sur dépôt git	
T11_3	Compléter le contrôleur (ajout/modification)	X		X	Code sur dépôt git	
T11_4	Intégration de la fonctionnalité (ajout/modification) à l'application	X		X	Code sur dépôt git	
T11_5	Tester l'intégration de la fonctionnalité		X		Rapport de test fonctionnel	
T11_6	Création de la vue de suppression d'une représentation		X		Code sur dépôt git	
T11_7	Compléter le contrôleur		X		Code sur dépôt git	
T11_8	Intégration de la fonctionnalité à l'application		X		Code sur dépôt git	
T11_9	Tester l'intégration de la fonctionnalité	X		X	Rapport de test fonctionnel	
<b>T12</b>	<b>Déploiement</b>					
T12_1	Déployer l'application sur le serveur de test de l'équipe	X	X			
T12_2	Tester l'application déployée			X	Rapport de test	

#### IV.2.5.2<sup>2</sup>Gestion de rôles d'utilisateurs

L'application devient une application extranet, utilisée par le personnel de la mairie, mais aussi des intervenants extérieurs.

On définit des types d'utilisateur (rôles) afin de mieux contrôler les droits de chacun.

Chaque utilisateur se voit affecter un rôle :

- Gestionnaire (Mairie) : accès à toutes les données en CRUD (\*).
- Établissement : accès à toutes les données en consultation uniquement, à ses offres en CRUD, à son profil en modification.
- Visiteur authentifié : accès à toutes les données en consultation seulement.
- Les habilitations sont fixées une fois pour toutes.

(\* ) *CRUD* : create, read, update, delete

#### Travail à faire :

- Tâche n° 11
  - apporter les modifications nécessaires au diagramme de classes ainsi qu'au MCD
  - compléter les couches métier et DAO et programmer les tests unitaires
  - modifier le contrôle de l'accès aux pages en mode connecté, pour vérifier l'habilitation en fonction des rôles ;
  - procéder aux tests fonctionnels.
- Tâche n° 12
  - déployer l'application sur le serveur de test de l'équipe
  - tester l'application déployée

### IV.3 Itération n°7 - livraison de l'application

	Tâche	E1	E2	E3	Production	Activité
	<b>Itération 7</b>	Livraison de l'application				
<b>T13</b>	Déploiement n° 2					
T13_1	Déployer l'application sur le serveur de production	X	X			
T13_2	Tester l'application déployée			X	Rapport de test	
<b>T14</b>	Clôture					
T14_1	Créer le planning réel à partir des cartes trello	X	X		Diagramme de Gantt (GanttProject)	
	<b>Évaluation 3 par les professeurs</b>				<b>Note 3</b>	

## V - Annexes

### V.1 Les lieux et les représentations, exemples

Extrait de la table Lieu :

id	nom	adr	capacite
1	SALLE DU PANIER FLEURI	Rue de Bonneville	450
2	LE CABARET	MAIRIE ANNEXE DE PARAMÉ, Place Georges COUDRAY	250
3	LE PARC DES CHENES	14 rue des chênes	2000
4	LE VILLAGE	Ecole LEGATELOIS, 25 rue Général de Castelnaud	500

Exemple d'affichage des représentations :

Accueil	Gestion établissements	Gestion types chambres	Offre hébergement	Attribution chambres
<b>Représentations</b>				
<b>Programme par jours</b>				
<b>11/07/2017</b>				
Lieu	Groupe	Heure Début	Heure Fin	
SALLE DU PANIER FLEURI	Panama Fuerte Raza	20:30	21:45	<a href="#">Modifier</a> <a href="#">Supprimer</a>
SALLE DU PANIER FLEURI	Le Ballet Rey	21:45	23:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE CABARET	Boxty	19:00	20:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE CABARET	Soig Siberil et Etienne Grandjean	20:30	21:30	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE CABARET	Le Bour-Bodros	21:45	23:15	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE PARC DES CHENES	Bagad Montfortais	11:00	12:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE PARC DES CHENES	Cercle Ar Vro Melenig	12:00	13:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
<b>12/07/2017</b>				
Lieu	Groupe	Heure Début	Heure Fin	
SALLE DU PANIER FLEURI	Aira da Pedra	20:30	22:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
SALLE DU PANIER FLEURI	The Jersey Caledonian Pipe Band	22:15	23:30	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE CABARET	Ensemble Kidra Budaya	20:00	23:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
<b>13/07/2017</b>				
Lieu	Groupe	Heure Début	Heure Fin	
SALLE DU PANIER FLEURI	Compagna Folklorica Camagua	20:30	22:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE CABARET	Jackie Molard Quartet	20:30	22:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
<b>14/07/2017</b>				
Lieu	Groupe	Heure Début	Heure Fin	
SALLE DU PANIER FLEURI	L'Alen	19:30	21:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
SALLE DU PANIER FLEURI	Deloubes Kévin	21:15	23:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE PARC DES CHENES	Groupe folklorique des Émirats	14:00	14:30	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE PARC DES CHENES	Groupe folklorique mexicain	14:30	15:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE PARC DES CHENES	Groupe folklorique de Panama	15:00	15:30	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE PARC DES CHENES	Groupe folklorique papou	15:30	16:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE PARC DES CHENES	Groupe folklorique turc	16:00	16:30	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE PARC DES CHENES	Groupe folklorique russe	16:30	17:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE VILLAGE	Vent de Noroïse	11:00	12:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE VILLAGE	Kan Tri	15:00	17:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE VILLAGE	Deomp	17:30	19:30	<a href="#">Modifier</a> <a href="#">Supprimer</a>
<b>15/07/2017</b>				
Lieu	Groupe	Heure Début	Heure Fin	
LE VILLAGE	Kerc'h Keltiek Roazhon	11:00	12:30	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE VILLAGE	Soeurs Chauvel	15:00	16:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>
LE VILLAGE	Savidan Launay	16:30	19:00	<a href="#">Modifier</a> <a href="#">Supprimer</a>

## V.2 Scénarios d'authentification

### Cas d'utilisation : Authentification

#### Scénario n°1 : cas nominal

Personne n'est connecté ; la page d'accueil est affichée en mode déconnecté : un bouton « Se connecter » est disponible, les liens des autres onglets ne sont pas fonctionnels

1. L'utilisateur clique sur le bouton « se connecter »	
	10. Le système affiche la page de connexion
L'utilisateur s'identifie (login et mot de passe) et clique sur le bouton «Valider»	
	4. Le système vérifie la correspondance login/mot de passe pour cet utilisateur 5. L'utilisateur est authentifié, le système affiche la page d'accueil avec des liens fonctionnels et le nom de l'utilisateur connecté + un bouton se déconnecter

#### Scénario n°2 : cas alternatif : échec de l'authentification

Le scénario reprend au point n°4 du scénario nominal

	4. Le système vérifie la correspondance login/mot de passe pour cet utilisateur 5. L'utilisateur n'est pas authentifié, le système affiche la page de connexion avec un message d'erreur
--	---

#### Scénario n°3 : cas alternatif : abandon

Le scénario reprend au point n°3 du scénario nominal

4. L'utilisateur clique sur le bouton « Annuler »	
	5. Le système affiche la page d'accueil en mode non connecté

### Cas d'utilisation : Déconnexion

#### Scénario n°1 : cas nominal

Une personne est connectée ;une page quelconque est affichée, un bouton « Se déconnecter » est disponible ; le nom de l'utilisateur connecté est affiché.

1. L'utilisateur clique sur le bouton « se déconnecter »	
	2. Le système affiche la page d'accueil en mode non connecté