

Cahier des charges - Projet Eirplan:

Description fonctionnelle :

Utilisation d'un écran tactile Kimex 55 pouces sous Android afin de fournir des informations géographiques durant les différents événements à l'école aux utilisateurs.

Objectif:

Développer une interface web afin de fournir de l'information géolocalisée aux visiteurs/étudiants (les bureaux, les amphis, les salles TD/TP, etc ...) ou durant les différents événements de l'école (Conférences, Forum Ingénib, Rencontre Métiers, etc...).

Les acteurs :

1. Les étudiants : viennent chercher les entreprises qui les intéressent, les domaines qui les intéressent, les technologies
2. Les entreprises : viennent exposer leurs offres de stage, d'emploi, leur domaine d'activité, communiquer sur leurs marques
3. Les organisateurs : faire en sorte que l'événement se déroule bien

Les fonctionnalités :

FP1 : Informer les visiteurs lors d'un événement

FP1.1. Fournir des informations géographiques sur l'événement

FP1.2. Montrer les thèmes abordés par l'événement

FP1.3. Mettre en valeur l'événement et les organisateurs avec un logo et une description

FP2 : Visualiser une carte géographique de l'événement

FP2.1. Visualiser l'information sur une carte

FP2.2. Visualiser les différents étages

FP3 : Permettre au visiteur de rechercher des informations

FC1 : Être lisible de nuit comme de jour

FC1.1 : Disposer d'un thème sombre pour la nuit

FC1.2 : Disposer d'un thème clair pour le jour

FC2 : S'adapter à différentes tailles et géométrie d'écran

FC3 : Disposer d'un backoffice hors afficheur pour alimenter Eirplan en informations

FC3.1. : Disposer d'un Mdp pour pouvoir modifier

FC3.2. : Disposer d'une liste contenant les différents plans de l'école et les différentes entreprises

FC3.3. : Possibilité de modifier et/ou ajouter des nouveau plans / nouvelles entreprises

FC4 : Disposer d'un arrêt et d'une mise en marche automatique à des horaires paramétrables

FC5 : Être 100% opensource et partagé

Etat de l'art :

Fonctionnel :

- Projets similaires : affichage dynamique, Dynamic Screen, propriétaire et opensource, ...

Dynamic Screen:

1. Diffusion sans interaction
2. Ce n'est pas un opensource
3. Nouveau projet avec un écran tactile

TODO : à compléter avec d'autres projets commerciaux ou opensource, avantages et inconvénients

Indoor3D:(<https://github.com/wolfwind521/indoor3D?fbclid=IwAR0cVikDyQQKU Xye8D5kVhOx7OBwxVknOMBhb86U1HflkPo2hdq4rn xv7XA>)

Bibliothèque JS permettant de visualiser une map 2D/3D:

1. Utilisation de Three.js
2. Langages de programmation utilisés: JS, HTML, CSS
3. +: Utilisation Simple
4. +: Opensource
5. +: Sélection de plusieurs étages et points d'intérêt

6. -: Ne supporte pas tout les browsers ex: Firefox
7. -: Code complexe à suivre

Interactive 3D Mall Map:

(<https://github.com/codrops/Interactive3DMallMap/blob/master/README.md>)

Une carte de centre commercial 3D interactif avec un bar de recherche et des tags pour chaque niveau.

1. Langages de programmation utilisés: JS, HTML, CSS
2. Utilisation de List JS pour la partie recherche
3. +: opensource: License MIT
4. +: bar de recherche avec triage
5. +: utilisation de SVG et intégration des paths dans le code
6. +: sélection de différents points d'intérêt et affichage d'une description de chaque point(Les tags géolocalisés)
7. +: Code facile à suivre
8. -: plan en 2D

Clavier: (<https://github.com/topics/kiosk?l=javascript&o=desc&s=updated>):

1. Opensource clavier en utilisant JS

Projets similaires: la plupart utilise Javascript + html + css

- Comment on se différencie et qu'est-ce qui est similaire ?

Technique :

- React :
 1. DOM virtuel
 2. Flux de données unidirectionnel
 3. Opensource Bibliothèques
- 4. Publication de code opensource légal
- 5. Peu de documentations en ligne
- React Native: JavaScript framework pour développer une application sous Android
- NuxtJS :
 1. VueJs framework
 2. opensource
 3. Création de SPA
 4. Création de site universel serveur/client
 5. -: peu de ressources sur Internet

Tutorial utilisation et intégration de mode de nuit/jour en ligne

- Node JS : 1. Opensource (MIT License)
2.
- NW js : 1. Ouvrir des sites web comme une application.
2. Application marche sur Windows, Mac OS et Linux
3. Accès aux fichiers locaux sur le desktop (avantage pour uploader les plans facilement)
4. Interaction avec le système via des APIs JavaScript et NodeJs
5. Compatible de travailler avec React et NodeJS
- 6. Alternatif pour Android?
- ThreeJs
- Unity 3D: À partir d'une image du plan, on crée un plan 3D
- List js : pour la partie recherche
 - Pas besoin de mot complet
 - Ordonne + recherche
 - HTML + JavaScript
- Inkscape : Manipuler le SVG : <https://inkscape.org/fr/>
- Lire les fichiers XML/SVG avec une bibliothèque JS à trouver
SVG.js: (<https://svgjs.com/docs/3.0/>)
 - 1. +: Petite taille(11K)
 - 2. +: supporte les animations
 - 3. +: supporte les transformations et le changement de couleurs
 - 4. +: supporte les texts paths et les paths incomplets
 - 5. +: opensource
- wordcloud2.js:(<https://github.com/timdream/wordcloud2.js>)
Bibliothèque pour la génération d'un nuage de mots
- D3-worldcloud: (<https://github.com/wvengen/d3-wordcloud>)
Bibliothèque pour la génération d'un nuage de mots

Conception :

Ecrans : dessiner les écrans frontoffice ET backoffice et détailler leurs interactions entre eux (flèches, commentaires, ...)

Cas d'utilisation :

CU1. Consulter le plan

Acteurs : les entreprises, les étudiants

Précondition : L'organisateur a enregistré un plan sur au moins 1 étage

Scenario :

- L'utilisateur clique sur un étage
- TODO AJouter le clic sur un tag géolocalisé

Postcondition : Il visualise les différents points d'intérêt de cet étage sur le plan

CU2. Chercher les informations géographiques

Acteurs: Les étudiants, les entreprises

Précondition: BDD avec les mots-clés possibles.

Scenario:

- L'utilisateur clique sur le bar recherche
- L'utilisateur tape un nom de tag (Pas besoin d'être complet)

Postcondition: Il visualise un nuage de mots clés liés à ce qu'il a cherché.

CU3. Présentation de l'événement

Acteurs: Les étudiants, Les entreprises

Précondition: Les organisateurs ont ajouté une présentation de l'événement et une liste des participants.

Scenario:

- L'utilisateur clique sur le point d'interrogation.

Postcondition: Il visualise une page avec la description de l'événement et une liste des entreprise participant.

CU4. Ajout de la présentation de l'événement

Acteurs: Les organisateurs

Préconditions: L'organisateur possède de la liste des participants.

Scenario:

- L'organisateur se logue sur le backoffice
- Il clique sur "Ajouter un événement"
- Il ajoute une description de l'événement
- Il ajoute le logo de l'événement
- Il ajoute le nom des entreprises participant + une description et leur domaine de travail

Postcondition: le fichier est enregistré et mis à jour directement sur le frontoffice

CU5. Alimenter Eirplan avec un nouveau plan

Acteur : les organisateurs

Précondition : L'organisateur a dessiné en SVG le plan d'un étage et placé des points comprenant les tags

Scenario :

- L'organisateur se logue sur le backoffice
- Il clique sur "Plan"
- Il sélectionne l'étage à modifier ou ajouter
- Il clique sur "Parcourir" pour envoyer son plan SVG

Postcondition :

- Si le fichier est invalide, il est rejeté avec une explication
- Si le fichier est valide, il est enregistré et mis à jour directement sur le frontoffice

Implémentation :

- Définition des langages de programmation à utiliser.
Proposition: Partie Front: React + ThreeJS partie 3D.
NuxtJs pour les modes jour/nuit
NW js pour l'application
Partie Back: MongoDB pour la BDD + NodeJs.
- Plan de l'école: Besoin de plan architecture des bâtiments

Idées d'amélioration:

Choisir le thème et les couleurs de l'interface. Proposition: Couleurs de logo de l'école avec un background blanc salé pour le mode light.

Mode sombre (Nuxt Js)

Animations

La bdd: les svgs

Les logos

table params(activer les modes)

Cherche le mode debug node