|  |
| --- |
| Projet I-Shoes |



Alex Barreira-videra – Scrum master

Romain Lenoir – Responsable développement

Sacha Jaccard – Responssable maquettes et documentation

Table des matières

[1 Analyse préliminaire 4](#_Toc130472212)

[1.1 Introduction 4](#_Toc130472213)

[1.2 Objectifs 4](#_Toc130472214)

[1.3 Planification 5](#_Toc130472215)

[2 Analyse / Conception 5](#_Toc130472216)

[2.1 Concept 5](#_Toc130472217)

[2.2 Stratégie de test 12](#_Toc130472218)

[2.3 Planification 12](#_Toc130472219)

[2.4 Dossier de conception 12](#_Toc130472220)

[3 Réalisation 13](#_Toc130472221)

[3.1 Dossier de réalisation 13](#_Toc130472222)

[3.2 Analyse des sprints 14](#_Toc130472223)

[3.3 Modifications 15](#_Toc130472224)

[3.4 Stratégies de test 16](#_Toc130472225)

[3.5 Erreurs restantes 16](#_Toc130472226)

[4 Conclusions 17](#_Toc130472227)

[5 Annexes 17](#_Toc130472228)

[5.1 Sources – Bibliographie 17](#_Toc130472229)

[5.2 Journal de travail 17](#_Toc130472230)

[5.3 Archives du projet 17](#_Toc130472231)

# Analyse préliminaire

## Introduction

I Shoes est un site d’e-commerce spécialisé dans la vente de chaussure. Il est réalisé dans le cadre d’un projet d’e-commerce pour le cours de Projet Web des deuxièmes années du CPNV. Ce site permettra aux utilisateurs d’entretenir un compte avec son propre panier et sa propre empreinte d’achat. Il sera ainsi possible de naviguer sur tout le site afin d’y rencontrer nos derniers articles. Un compte administrateur est également propice où l’utilisateur pourra créer un nouvel article, y rajouter son stock et à en plus de cela accès à un historique d’achat. Une vue ‘utilisateur’ est également mis pour l’administrateur du site.

## Objectifs

L’objectif du projet est d’avoir les options ci-dessous :

* Pouvoir accéder au site.
* Pouvoir créer un nouveau compte utilisateur.
* Pouvoir se connecter aux comptes utilisateurs précédemment créer.
* Avoir un page de descriptif entre chaque article.
* Concevoir une page administrateur pour l’administration des articles et d’informations sur le site.
* Avoir un historique d’achat des utilisateurs pour l’administrateur lui-même.

## Planification

La planification du projet a été conçue et planifié avec l’aide du site iceScrum avec l’URL ci-dessous:

*https://icescrum.cpnv.ch/p/PROJWEB202/#/project*

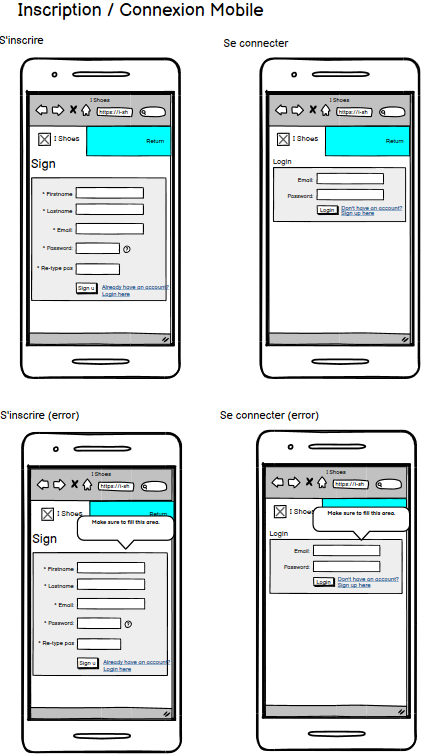
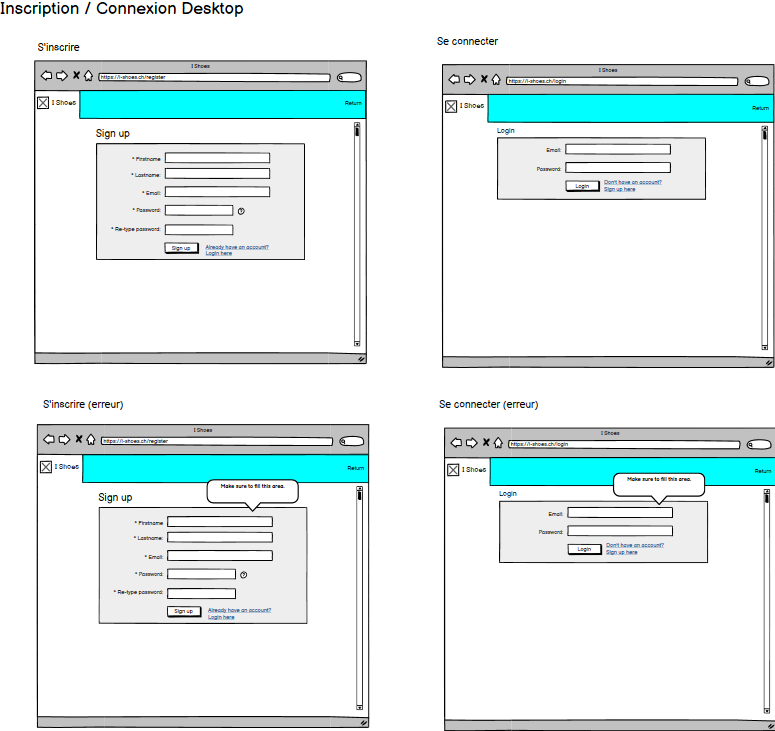
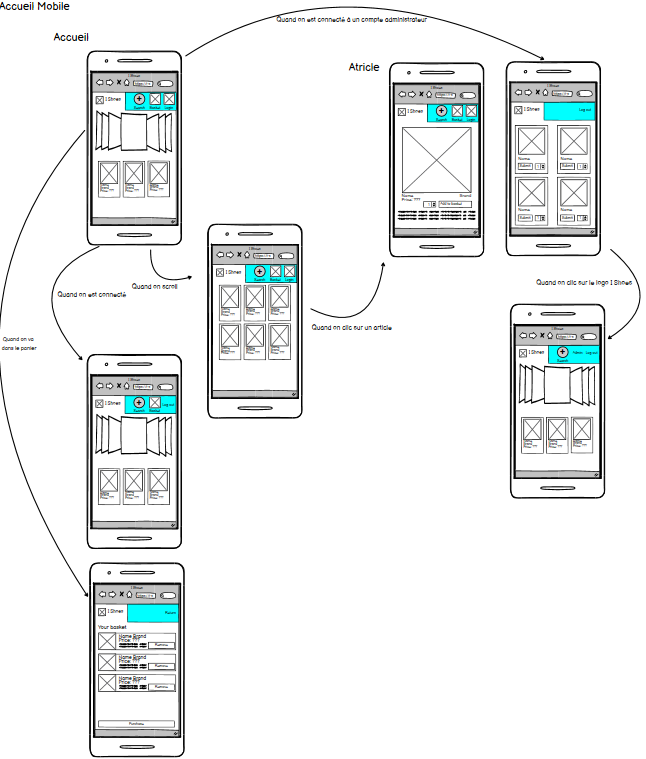
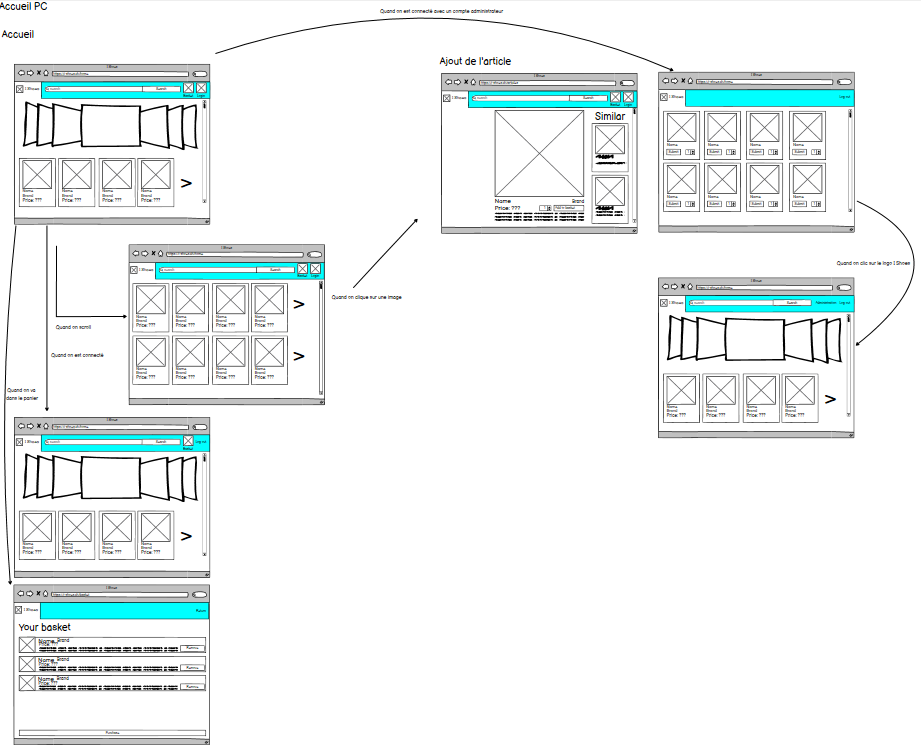
# Analyse / Conception

## Concept

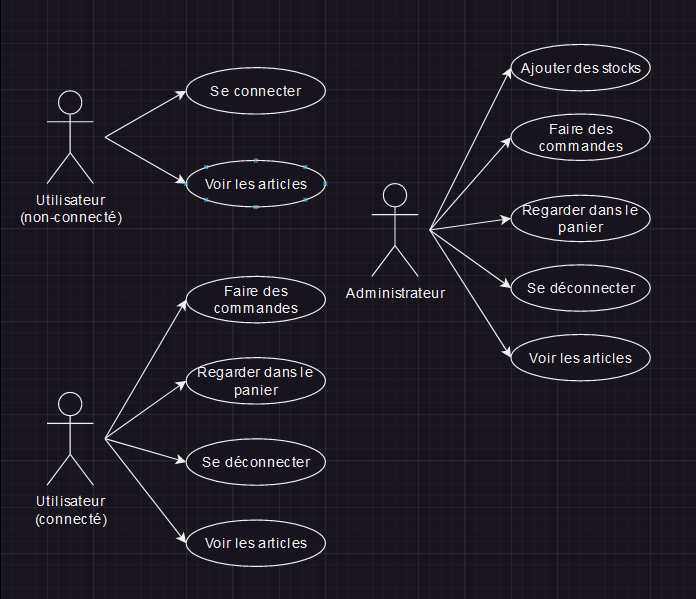
L’objectif du projet est d’avoir les options ci-dessous :

* Pouvoir accéder au site.
* Pouvoir créer un nouveau compte utilisateur.
* Pouvoir se connecter aux comptes utilisateurs précédemment créer.
* Avoir un page de descriptif entre chaque article.
* Concevoir une page administrateur pour l’administration des articles et d’informations sur le site.
* Avoir un historique d’achat des utilisateurs pour l’administrateur lui-même.
* Avoir un message montrant le contenu de la commande après en avoir passé une.

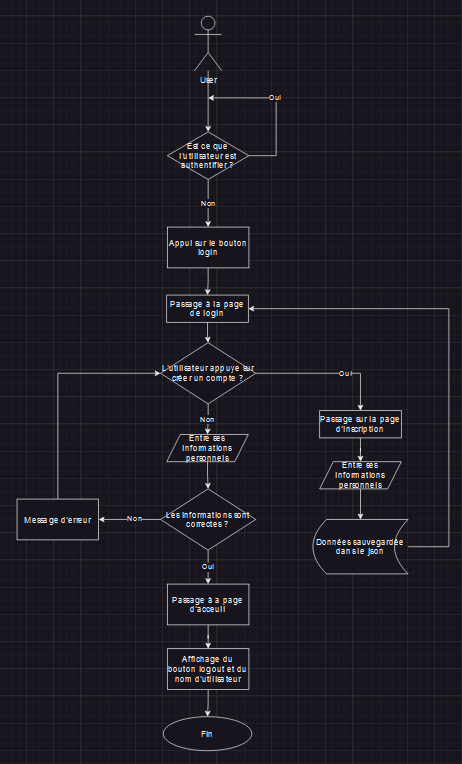
**Wireframes**



**Use case**



**Diagramme de flux**



## Stratégie de test

Description de la stratégie globale de test :

* Si un utilisateur lambda clique sur un article, l’utilisateur se retrouve sur la page de l’article spécifié.
* Si l’utilisateur met son article demandé sous son panier, il se glisse sous celle-ci.
* Si l’utilisateur envoie son panier, il est ensuite en cours de payement et l’utilisateur se voit redirigé sur une page d’achat.
* Si l’utilisateur n’inscrit pas les bons logges, un message d’erreur s’affiche.
* Si l’utilisateur créer un compte utilisateur qui n’est pas unitaire, un message d’erreur s’affiche.
* Si l’utilisateur essaie d’ajouter son article dans son panier alors qu’il n’est pas connecté par son compte, un message d’erreur s’affiche et l’utilisateur est redirigé sur la page php ‘login’.
* Si l’utilisateur se voit malencontreusement à mettre dans son panier un article qui n’a pas de stock ou qu’il en demande plus que cet article comporte, un message d’erreur s’affiche.
* SI l’administrateur créait un article avec une photo non-négligeable, un message d’erreur s’affiche.
* Si l’administrateur est connecté, il aura accès à la page d’administration du site. Il pourra créer des articles et y rajouter leur stock.
* Si l’utilisateur décide d’enlever son article dans son panier, l’article en question se fait enlever de son article.
* Si l’administrateur décide d’enlever un article créé, l’article en question se fait enlever de la base de données des articles.

## Planification

Révision de la planification initiale du projet :

* Le projet a commencé le 02.02.2023 et terminera durant la semaine du 4 avril 2023.
* Alex est devenu Scrum master.
* Romain est responsable développement et a aidé pour la création et finition de la documentation.
* Sacha est responsable des maquettes à concevoir et de la documentation du projet.

## Dossier de conception

Fournir tous les documents de conception :

* Pour ce dit projet, le système d’exploitation Windows 10 est utilisé.
* Pour les maquettes du projet, ‘Balsamiq’ est le logiciel utilisé.
* Pour la réalisation de diagramme de flux, le logiciel et site internet ‘Draw.io’ est utilisé
* Pour la réalisation des journaux de travail, Excel est le logiciel utilisé.
* Pour la réalisation et structure du code, le php et l’interpréteur php est utilisé.
* Pour la documentation et notes du projet, Word est le logiciel utilisé.
* Pour l’exportation des différentes maquettes, du Wireframe et des user-cases, le format PDF est utilisé.
* Pour la structure des bases de données, nous avions sobrement voulu que cette structure soit simple et qu’il soit également facile à comprendre.

# Réalisation

## Dossier de réalisation

Réalisation « physique » du projet

Répertoires

* .idea
  + .gitignore
  + .name
  + I\_Shoes.iml
  + modules.xml
  + php.xml
  + vcs.xml
  + watcherTasks.xml
* Code
  + .idea
    - .name
    - Code.iml
    - modules.xml
    - vcs.xml
    - workspace.xml
  + controllers
    - navigation.php
  + data
    - dataArticles.json
    - dataBasket.json
    - dataPurchases.json
    - dataUser.json
  + media
    - img
      * articles
        + (Source images des articles)
      * (Source images du site)
    - scripts
      * slider.js
    - stylesheet
      * style.css
  + models
    - model.php
  + temp
    - phpinfos.php
  + views
    - admin.php
    - basket.php
    - historic.php
    - home.php
    - layout.php
    - layout\_form.php
    - login.php
    - lost.php
    - main.php
    - new\_user.php
    - show\_article.php
    - TDC\_admin.php
  + index.php
* Documentation
  + .$Diagramme de flux I-Shoes.drawio.bkp
  + Cahier des charges Projet I Shoes.docx
  + Documentation I Shoes.docx
  + Journal\_de\_travail
    - JournalTravailAlex
      * Journal\_de\_travail\_I\_Shoes\_ALEX.xlsx
    - Romain Lenoir
      * Journal\_de\_travail\_I\_Shoes.xlsx
    - Sacha Jaccard
      * Journal de travail Sacha.xlsx
  + Maquettes
    - .$Use case.drawio.bkp
    - AccueilMaquette.bmpr
    - Diagramme de flux I-Shoes.drawio
    - I Shoes Wireframe.bmpr
    - Use case.drawio
    - Use case.drawio.pdf
    - Wireframe.pdf
  + Cahier des charges Projet I Shoes.pdf
* php
  + (Fichiers de configuration de l’interpréteur PHP)
* .gitattributes
* README.md

Versions des systèmes d’exploitation

* Balsamiq Wireframes version: 4.7.1
* Excel version: 2016
* GitHub Desktop version : …
* PhpStorm 2022.2.1 version: PHP: 5.6
* Word version : 2016

Programmation et scripts

* …

## Analyse des sprints

Sprint 1

* …

Sprint 2

* …

Sprint 3

* …

Sprint 4

* Création de la page d’accueil
* Création du model de conception MVC
  + Création du Model dans le code
  + Création du View dans le code

Sprint 5

* …

Sprint 6

* …

Sprint 7

* …

Sprint 8

* …

Sprint 9

* …

## Modifications

* Page d’administration différente du Wireframe

## Stratégies de test

* Si un utilisateur lambda clique sur un article, l’utilisateur se retrouve sur la page de l’article spécifié.
* Si l’utilisateur met son article demandé sous son panier, il se glisse sous celle-ci.
* Si l’utilisateur envoie son panier, il est ensuite en cours de payement et l’utilisateur se voit redirigé sur une page d’achat.
* Si l’utilisateur n’inscrit pas les bons logges, un message d’erreur s’affiche.
* Si l’utilisateur créer un compte utilisateur qui n’est pas unitaire, un message d’erreur s’affiche.
* Si l’utilisateur essaie d’ajouter son article dans son panier alors qu’il n’est pas connecté par son compte, un message d’erreur s’affiche et l’utilisateur est redirigé sur la page php ‘login’.
* Si l’utilisateur se voit malencontreusement à mettre dans son panier un article qui n’a pas de stock ou qu’il en demande plus que cet article comporte, un message d’erreur s’affiche.
* Si l’administrateur créer un article avec une photo non-négligeable, un message d’erreur s’affiche.
* Si l’administrateur est connecté, il aura accès à la page d’administration du site. Il pourra créer des articles et y rajouter leur stock.
* Tentative d’ajout dans le panier de plusieurs articles avec des valeurs basses (de 1 à 9) négatives (de -1 à -999) et dans des valeurs exagéré (9999999). Puis regarder ce que ça donne pour l’achat des articles. Pour voir si les valeurs sont toute acceptées sans créer de problème et voir si le nombre d’achat d’article ne dépasse pas le stock existant.
* Si l’utilisateur quitte l’historique de la commande d’article et repasse commande l’historique sera mis à jour (remplaçant l’ancien).

## Erreurs restantes

…

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

# Conclusions

Pour conclure la réalisation du projet, nous avions remplit tous nos objectifs. Pour ce qui sera des problèmes rencontrés, nous avions eu certains problèmes avec la suppression et diminution en ce qui concerne les divers articles. Je pense, plus précisément, à l’utilisateur qui enlève de lui-même un stock d’un article avant de l’avoir acheté. Également, je pense aux problèmes que nous avions eu en créant un fichier .json pour le panier et un fichier .json pour les articles achetés. Le problème étant de bien administrer les deux en même temps, vu que l’un est aussi dépendant avec l’autre que son prochain => Ex : quand un utilisateur achète l’article en question, il est déjà inscrit sous sa page .json attribuée avant d’être supprimé sur la page de panier. A part cela, nous avions eu de grands espoirs pour ce projet et avions acquis des compétences au niveau de la compréhension du code et de réalisation de projet grâce à ceci.

# Annexes

## Sources – Bibliographie

Romain Lenoir =>

<https://forums.commentcamarche.net/forum/affich-9781202-css-force-retour-a-la-ligne>

<https://www.w3schools.com/php/func_array_splice.asp>

<https://stackoverflow.com/questions/11463581/how-to-use-unlink-function>

Sacha Jaccard =>

<https://app.diagrams.net/>

Alex Barreira-videira =>

…

## Journal de travail

## 

…

## Archives du projet

Toute les archives du projet ont été sauvegardé avec l’aide de Github avec l’URL ci-dessous:

<https://github.com/romain-lnr/I_Shoes>