

[**1. Introduction**](#_h77d0uz7qw9f) **3**

[**2. Contenu**](#_h77d0uz7qw9f) **3**

[2.1 Sujet](#_s1pk2ic2yo5h) 3

[2.2 But](#_3yzo3ia4dpmu) 3

[2.3 Spécifications](#_5o0alrx6udbx) 3

[2.4 Environnement de développement](#_kgejjquuz93v) 3

[2.5 Livrables](#_5o0alrx6udbx) 3

[**3. Analyse fonctionnelle**](#_h77d0uz7qw9f) **3**

[3.0 Profils utilisateur](#_415cnrw4hsfo) 3

[3.0Bis Les plans de navigation](#_1b7xncd2ipeh) 4

[3.0Bis.1 Visiteur(site web)](#_ojsm98f42kb1) 4

[3.0Bis.2 Utilisateur(site web)](#_5vyv791bod4j) 4

[3.0Bis.3 Visiteur(Application Android)](#_ix4hbhya0hb9) 4

[3.1 Partie visiteur(site web)](#_p1vdypews59) 5

[3.1.1 Login](#_ky2ilpfglydi) 5

[3.1.2 Register](#_lrdt5mz4x1q8) 6

[3.2 Partie utilisateur(site web)](#_mz0dq74lx1kf) 7

[3.2.1 Nav](#_byh3hgszwzao) 7

[3.2.2 Dashboard](#_ezkhhyuie1pw) 8

[3.2.3 Create event](#_oepwza1q8auq) 9

[3.2.4 Edit event](#_s9h1807gv7ih) 9

[3.2.5 Manage event(Not started yet state)](#_i23wtmgerkgj) 10

[3.2.6 Manage event(In progress state)](#_oyedvfm8p66z) 11

[3.2.7 Manage event(Past state)](#_fr8pnfun5nbu) 12

[3.3 Partie visiteur(Application Android)](#_cdth4evve550) 13

[3.3.1 Show Current/Coming events](#_d2uiv0dvnb97) 13

[3.3.2 Show Old events](#_wctz10r373d8) 14

[3.3.3 Event details](#_8sjtehl2f6p9) 15

[**4. Analyse organique**](#_h77d0uz7qw9f) **16**

[Manager utilisés](#_f79v8vy247qj) 16

[UserManager](#_cb16z9ry6yqh) 16

[Connect](#_6wa4fjliunim) 17

[**5. Tests**](#_h77d0uz7qw9f) **17**

[**6. Planning prévisionnel**](#_h77d0uz7qw9f) **17**

[**7. Planning effectif**](#_h77d0uz7qw9f) **17**

[**8. Conclusion**](#_h77d0uz7qw9f) **17**

[**9. Glossaire**](#_h77d0uz7qw9f) **17**

# 1. Introduction

Pour le Travail Pratique Individuel (TPI), nous avons connu des changements concernant son déroulement, ce TPI a en effet été réalisé à 100% en télétravail suite aux mesures prise suite à l’épidémie de COVID-19.

Mon chef de projet pour ce travail est M. Aigroz. La personne m’ayant fourni le travail à réaliser est M. Court, responsable TPI.

# 2. Contenu

## 2.1 Sujet

Création d’un site web de création/suppression/gestion d’événements ainsi qu’une application mobile Android de consultation d’événement en temps réel.  
Le projet se nomme Live events(Trad: Événements en direct).

## 2.2 But

Le but étant que des utilisateurs enregistrés puisse créer/supprimer/modifier des événements et gérer ces derniers en les alimentant de messages(flux) visibles en temps réel par des utilisateur non-authentifiés depuis une application android.

## 2.3 Spécifications

## 2.4 Environnement de développement

* Un PC standard école, 2 écrans
* Windows 10
* Apache + module php,MySQL Server
* MySQL Workbench
* Visual Studio Code + extensions
* Visual Studio 2019 community edition + Xamarin
* Suite Office
* 1 émulateur Android

## 2.5 Livrables

* Planning
* Documentation technique
* Le résumé du TPI
* Manuel utilisateur
* Journal de bord

# 3. Analyse fonctionnelle

## 3.0 Profils utilisateur

Le site web gère 2 profils de personne : visiteur et utilisateur.

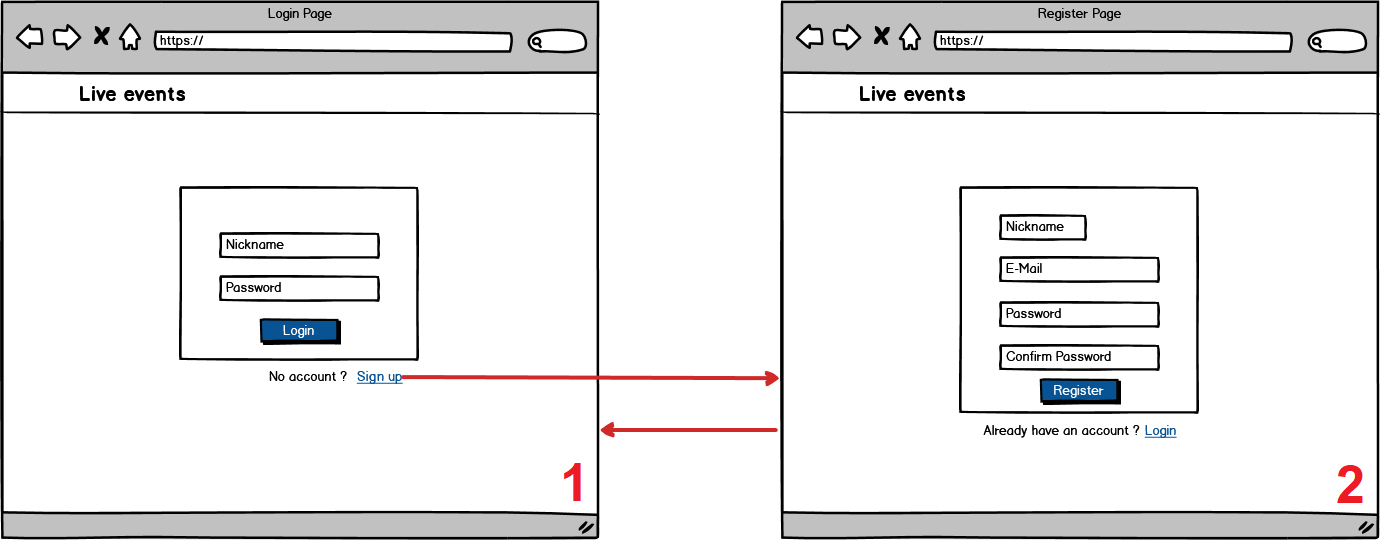
L’application mobile ne gère qu’un type de profil : visiteur

Chacun de ces profils donne accès à des fonctionnalités qui seront décrites ci-dessous.

## 3.0Bis Les plans de navigation

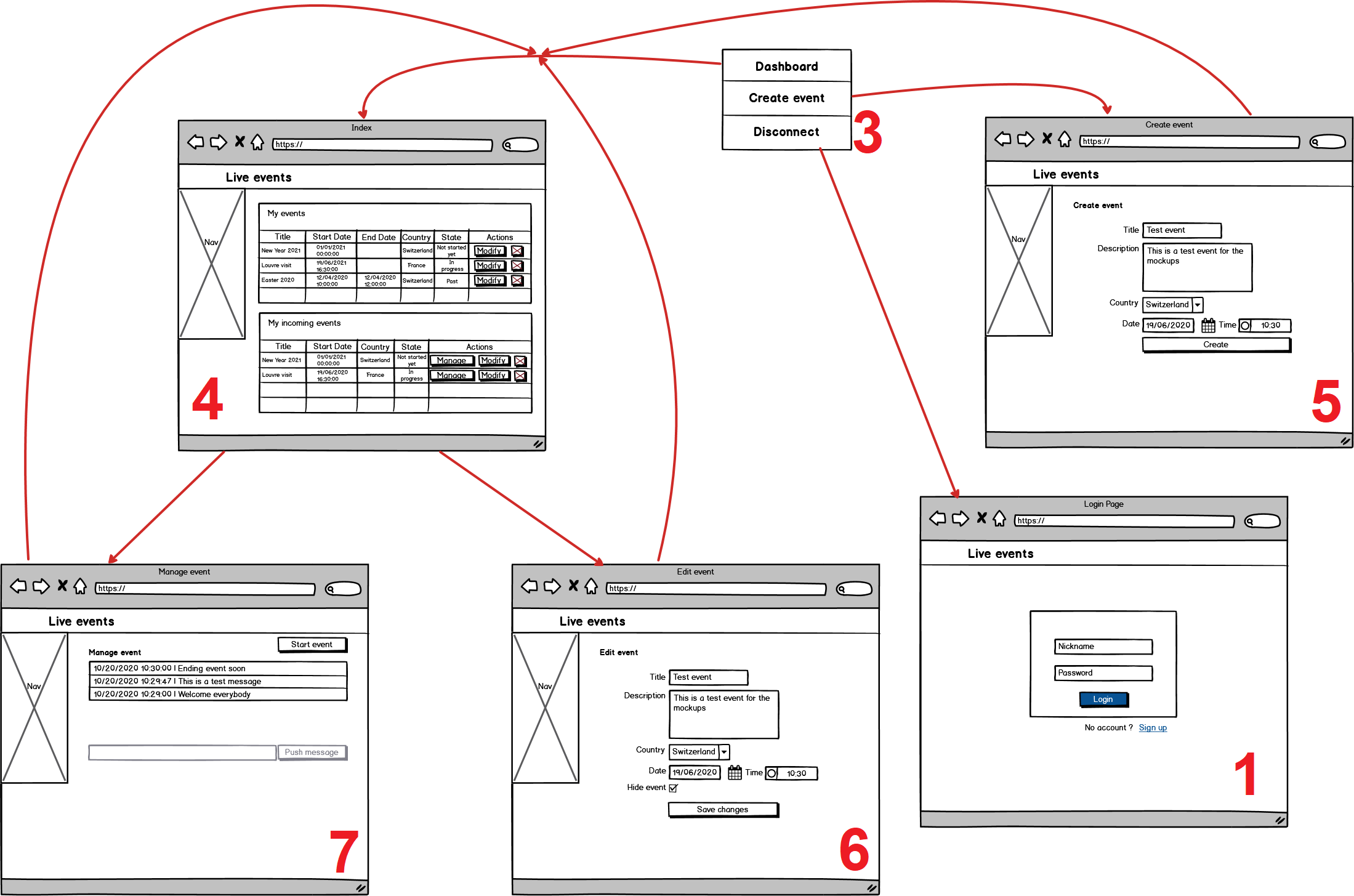
### 3.0Bis.1 Visiteur(site web)

Le visiteur ne peut que se connecter ou se créer un compte à l’aide des pages login et register.



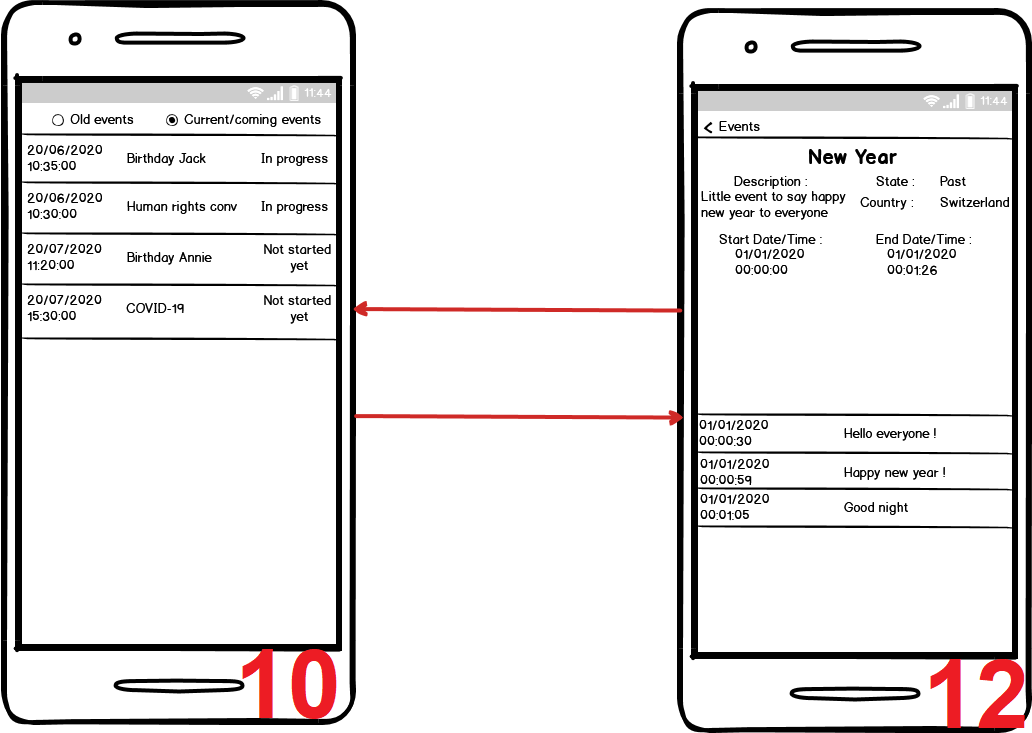
### 3.0Bis.2 Utilisateur(site web)

L’utilisateur a accès au tableau de bord, à la création, modification, gestion d’un événement.  
Lorsque l’utilisateur se déconnecte il est redirigé vers la page de connexion et repasse en profil visiteur.



### 3.0Bis.3 Visiteur(Application Android)

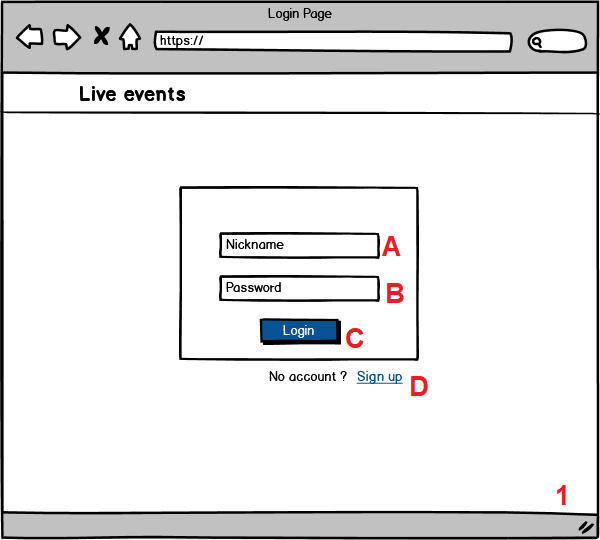
Le visiteur arrive sur la page de consultation de ses événements(figure 10) il peut accéder aux détails d’un événement en cliquant dessus, il arrive sur la figure 12 d’où il peut retourner sur la figure 10 en cliquant sur le bouton “Events”.



## 3.1 Partie visiteur(site web)

### 3.1.1 Login

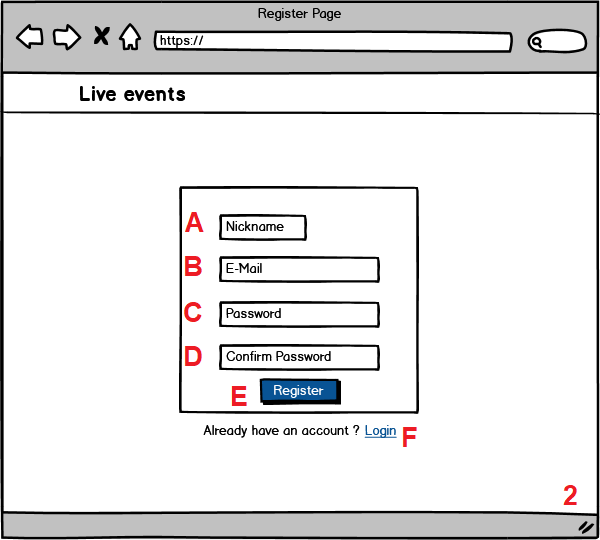
La page permettant à un utilisateur de se connecter avec ses identifiants, s’il n’a pas de compte il peut toujours cliquer sur le lien “Sign up” afin d’être rediriger vers la page de création de compte.



1. Le champ “Nickname” doit être rempli avec le pseudo de l’utilisateur
2. Le champ “Password” doit être rempli avec le mot de passe de l’utilisateur
3. Le bouton “Login” permet de confirmer la connexion à l’aide des information renseignées plus haut
4. Le lien “Sign up” redirige vers la page register.php permettant de se créer un compte utilisateur

### 3.1.2 Register

La page permettant de de se créer un compte utilisateur afin d'accéder au reste des fonctionnalités du site. Si l’utilisateur possède un compte et veut directement se connecter il peut cliquer sur le lien “Login” afin d’être redirigé vers la page de connexion.

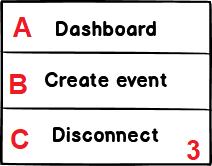


1. Le champ “Nickname” doit être rempli avec le pseudo souhaité par l’utilisateur
2. Le champ “E-mail” doit être rempli avec l’e-mail de l’utilisateur
3. Le champ “Password” doit être rempli avec le mot de passe souhaité par l’utilisateur
4. Le champ “Confirm Password” doit être rempli avec le même mot de passe que le point C afin de confirmer le mot de passe
5. Le bouton “Register” permet de confirmer la création du compte en envoyant les données entrées plus haut
6. Le lien “Login” permet à l’utilisateur d'accéder à la page login.php afin de se connecter si il a déjà un compte

## 3.2 Partie utilisateur(site web)

### 3.2.1 Nav

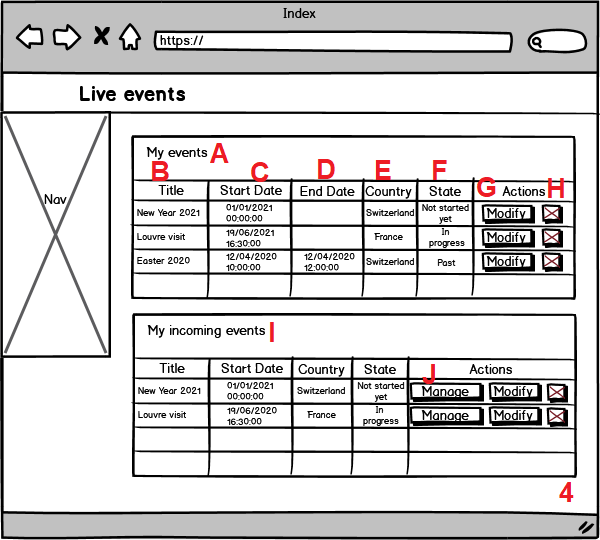
Le menu de navigation entre les différentes page du site web, il n’est présent que lorsque l’utilisateur est connecté.



1. Bouton redirigeant vers la page du tableau de bord affichant les événements de l’utilisateur
2. Bouton redirigeant vers la page de création d’événement
3. Bouton permettant de se déconnecter de la session en cours

### 3.2.2 Dashboard

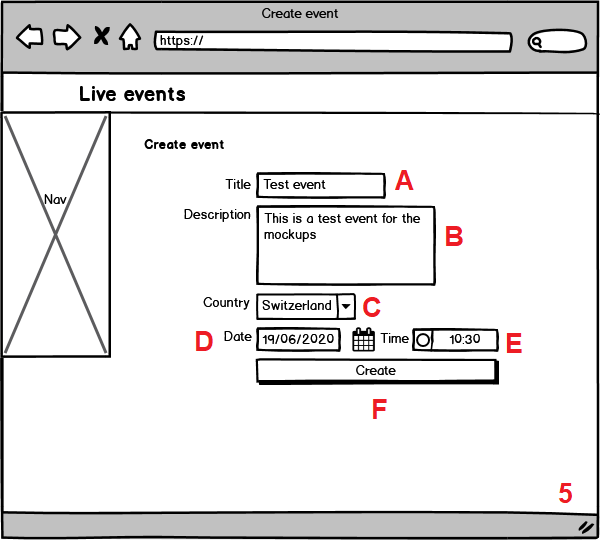
La page principale du site une fois connecté, elle permet de consulter les différents types d’événement que l’utilisateur a créé, il peut effectuer diverse action sur ces derniers : supprimer, accéder à la page de modification ou de gestion d’un événement.



1. La section “My events” regroupe tous les événements créés par l’utilisateur
2. La colonne “Title” renseigne le titre de chaque événement
3. La colonne “Start Date” renseigne la date et l’heure de début de chaque événement
4. La colonne “End date” renseigne la date et l’heure de fin de chaque événement si ils en ont une
5. La colonne “Country” renseigne le pays attribué à chaque événement
6. La colonne “State” renseigne l’état actuel de chaque événement, il peut avoir 3 états différents : “Not started yet”(Pas encore démarré), “In progress”(En cours), “Past”(Passé)
7. Le bouton modify redirige vers la page edit.php permettant de modifier les information d’un événement
8. Le bouton X permet de supprimer l’événement
9. La section “My incoming events” regroupe les événements en cours et à venir
10. Le bouton “Manage” redirige vers la page permettant de démarrer un événement et d’ajouter des infos au flux de ce dernier

### 3.2.3 Create event

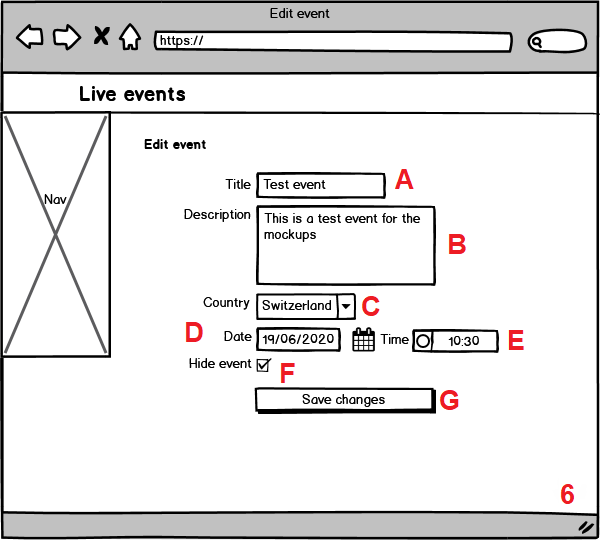
La page de création d’un événement, l’utilisateur y spécifie les diverse informations concernant l’événement et le créé.



1. Le champ “Title” doit être rempli avec le titre que l’utilisateur souhaite donner à son événement
2. Le champ “Description” doit être rempli avec la description de l’événement
3. Le dropdown “Country” permet de sélectionner le pays de l’événement
4. Le Datepicker “Date” permet de choisir la date de début de l’événement
5. Le Timepicker “Time” permet de choisir l’heure à laquelle l’événement commencera
6. Le bouton “Create” permet de confirmer l’envoi du formulaire rempli afin de créer l’événement

### 3.2.4 Edit event

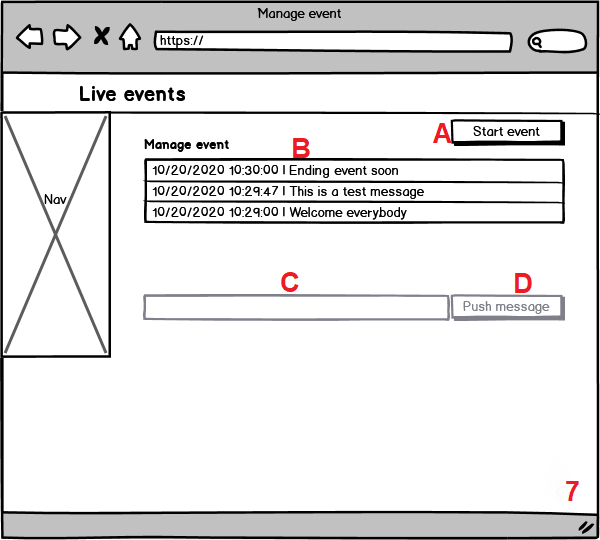
La page de modification d’un événement, l’utilisateur y spécifie les changements et valide le tout afin de mettre à jour les informations de l’événement.



1. Le champ “Title” doit être rempli avec le nouveau titre que l’utilisateur souhaite donner à son événement
2. Le champ “Description” doit être rempli avec la nouvelle description de l’événement
3. Le dropdown “Country” permet de sélectionner le nouveau pays de l’événement
4. Le Datepicker “Date” permet de choisir la nouvelle date de début de l’événement
5. Le Timepicker “Time” permet de choisir la nouvelle heure à laquelle l’événement commencera
6. La checkbox “Hide event” peut être cochée ou décochée afin de cacher ou rendre visible l’événement pour les utilisateurs consultant les événements sur leur smartphone
7. Le bouton “Save changes” permet de confirmer l’envoi du formulaire rempli afin de mettre à jour les informations de l’événement

### 3.2.5 Manage event(Not started yet state)

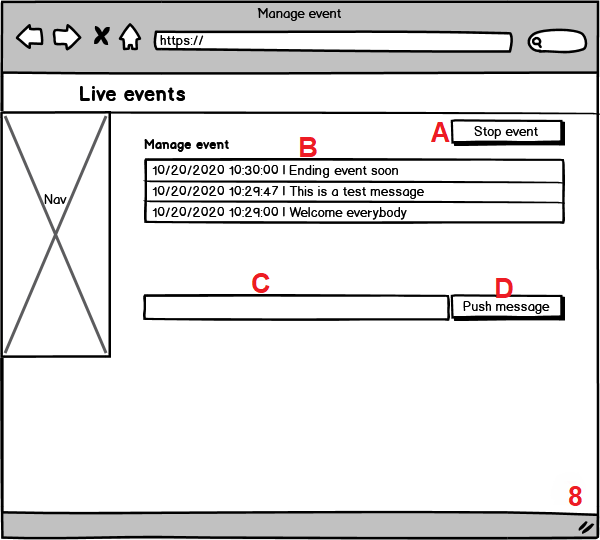
La page de gestion d’un événement, on y retrouve le flux de message de l’événement ainsi que la possibilité de commencer l’événement, dans cet état la page attend que l’utilisateur démarre l’événement avant de le laisser ajouter un message au flux de l’événement.



1. Le bouton “Start event” permet de démarrer l’événement, il passe son statut à : ”In progress”(En cours)
2. La section “Manage event” affiche en temps réel les informations du flux de l’événement sous la forme : Date Heure | Message
3. Champ permettant de rentrer un nouveau message à envoyer au flux de l’événement  
   il est à l’état “disabled” lorsque l’événement n’a pas été lancé avec le composant “Start event”(A).
4. Le bouton “Push message” permet de confirmer l’envoi du message entré dans le composant C afin d’ajouter un message au flux de l’événement, il est à l’état “disabled” lorsque l’événement n’a pas été lancé avec le composant “Start event”(A)

### 3.2.6 Manage event(In progress state)

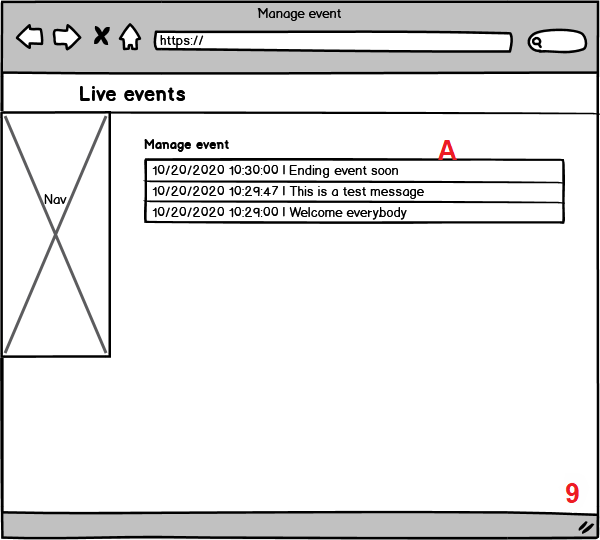
La page de gestion d’un événement, on y retrouve le flux de message de l’événement ainsi que la possibilité d’arrêter l’événement ou de rajouter un message au flux.



1. Le bouton “Stop event” permet d’arrêter l’événement, il passe son statut à : ”Past”(Passé)
2. La section “Manage event” affiche en temps réel les informations du flux de l’événement sous la forme : Date Heure | Message
3. Champ permettant de rentrer un nouveau message à envoyer au flux de l’événement
4. Le bouton “Push message” permet de confirmer l’envoi du message entré dans le composant C afin d’ajouter un message au flux de l’événement

### 3.2.7 Manage event(Past state)

La page de gestion d’un événement, on y retrouve le flux de message de l’événement passé on ne peut ni redémarrer l’événement ni y ajouter de nouveaux message au flux.



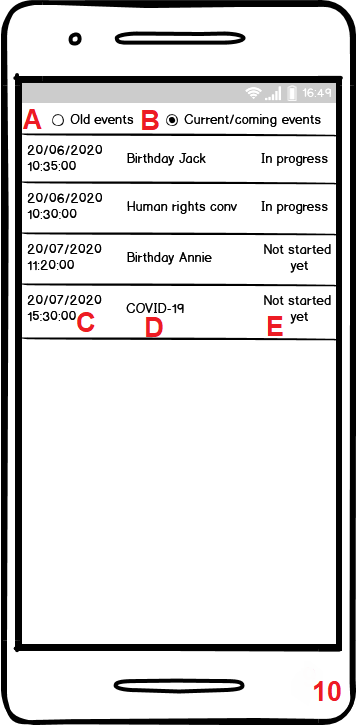
1. Affichage des messages passés du flux de l’événement sous la forme :  
   Date Heure | Message

## 3.3 Partie visiteur(Application Android)

### 3.3.1 Show Current/Coming events

L’écran principal de l’application android, accessible à tous, il permet de consulter tous les événements en cours ou futur, en jouant avec les radiobutton en haut de l’écran on peut remplacer les événements en cours/futurs par une liste des événements terminés.

Si l’on souhaite consulter le flux ou les informations concernants un événement il suffit de cliquer dessus dans la liste.

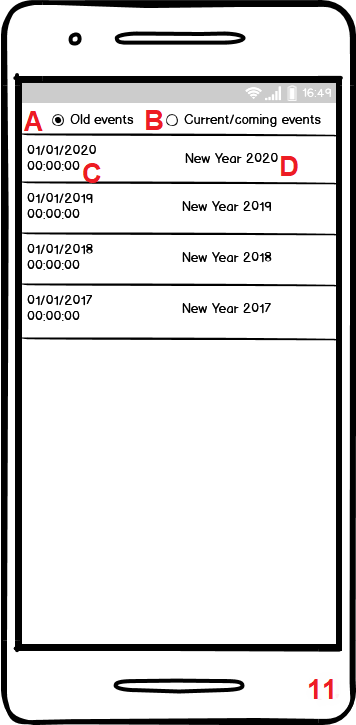


1. Le radiobutton “Old events” permet d’afficher tous les événements passés, il est lié au radiobutton “Current/coming events”(Figure B)
2. Le radiobutton “Current/coming events” permet d’afficher tous les événements en cours ou à venir, il est lié au radiobutton “Old events”(Figure A)
3. Label renseignant l’heure et la date de début de l’événement
4. Label renseignant le titre de l’événement
5. Label renseignant le statut de l’événement

### 3.3.2 Show Old events

L’écran principal de l’application android, accessible à tous, il permet de consulter tous les événements terminés, en jouant avec les radiobutton en haut de l’écran on peut remplacer les événements en terminés par une liste des événements en cours/futurs.

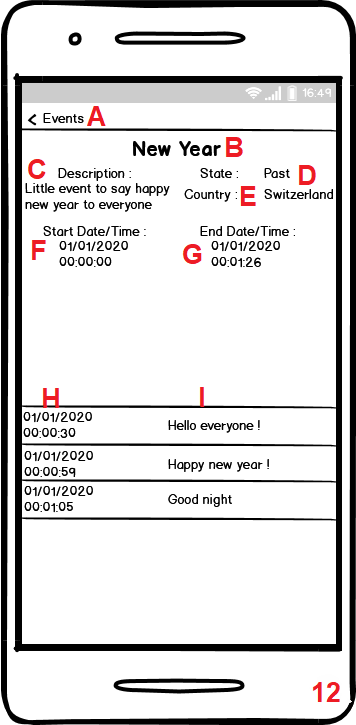
Si l’on souhaite consulter le flux ou les informations concernants un événement il suffit de cliquer dessus dans la liste.



1. Le radiobutton “Old events” permet d’afficher tous les événements passés, il est lié au radiobutton “Current/coming events”(Figure B)
2. Le radiobutton “Current/coming events” permet d’afficher tous les événements en cours ou à venir, il est lié au radiobutton “Old events”(Figure A)
3. Label renseignant l’heure et la date de début de l’événement
4. Label renseignant le titre de l’événement

### 3.3.3 Event details

On retrouve sur cet écran toutes les informations relative à l’événement sélectionné, on peut également consulter le flux de message d’un événement au bas de l’écran.



1. Le bouton “Events” permet de retourner à la page de consultation des événements
2. Label affichant le titre de l’événement sélectionné
3. Affichage de la description de l’événement sélectionné
4. Affichage du statut de l’événement sélectionné
5. Affichage du pays de l’événement sélectionné
6. Affichage de la date et de l’heure de début de l’événement sélectionné
7. Affichage de la date et de l’heure de fin de l’événement sélectionné si il en possède une
8. Affichage de la date et de l’heure des messages du flux de l’événement sélectionné
9. Affichage des messages du flux de l’événement sélectionné

# 4. Analyse organique

## 4.1 Manager utilisés

### 4.1.1 UserManager

Classe traitant les données utilisateur

#### connect

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| username  password | string  string  boolean | true si les 2 infos sont valide, false autrement |

#### 

#### userExist

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| username | string  boolean | true si le nom d’utilisateur existe sinon false |

#### createUser

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| user | Objet User  string | retourne le token de confirmation de l’utilisateur, retourne une chaîne vide sinon |

#### validateUser

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| token | string  boolean | true si l’utilisateur est validé sinon false |

### 4.1.2 SessionManager

#### addSession

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| index  value | string  - | - |

#### destroySession

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| - | - | - |

### 4.1.3 EmailManager

#### initMailer

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| - | - | - |

#### 

#### sendEmail

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| to  subject  text | string  string  string  boolean | true si le mail a été envoyé false autrement |

### 4.1.4 EventManager

#### getAllEvents

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| - | -  string | Retourne les événements au format JSON |

#### getUserEvents

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| username | string  string | Retourne les événements au format JSON |

#### createEvent

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| username  title  description  country  date  time | string  string  string  string  date  time  boolean | Retourne true si l’événement a pu être créé sinon false |

#### updateEvent

PARAMÈTRE(S) TYPE Valeur retournée

|  |  |  |
| --- | --- | --- |
| eventId  title  description  country  date  time  isVisible | int  string  string  string  date  time  boolean  boolean | Retourne true si l’événement a pu être mis à jour sinon false |

### 4.1.5 CountryManager

# 5. Tests

# 6. Planning prévisionnel

# 7. Planning effectif

# 8. Conclusion

# 9. Glossaire