

Cloud-
Computing-
Google-App-
Engine

2015

Ce TP a pour objectif de maîtriser les outils de développement fournis par le Google App Engine et de comprendre et maîtriser les contraintes liées à la réalisation d'une application Cloud

Réalisation
d'une
application
Cloud

Table des matières

Table des matières	2
1. Introduction.....	3
2. Pages principales.....	4
2.1. Page d'accueil.....	4
2.2. Page de recherche.....	4
3. Page d'ajout de plan d'entraînement	6
3.1. Page Web.....	6
3.2. Ajout du plan d'entraînement.....	6
4. Page de resultat de recherche	7
4.1. Page Web.....	7
4.2. Search function.....	7
4.3. News.....	8
5. Page de travail : détail d'un plan d'entraînement.....	8
5.1. Page Web.....	8
5.2. Comportement.....	9
6. Visualisation des données personnels	9
6.1. Page Web.....	9
7. Communication avec un coach (optionnel)	10
7.1. Page Web.....	10
7.2. Fonctionnement.....	11

1. Introduction

L'objectif de ce TP est d'appréhender la programmation Cloud Computing en mettant en œuvre une application Google App Engine permettant de valider/créer des plans d'entraînement, enregistrer et récupérer ces données en base, la recherche d'information dans une base, l'optimisation d'enregistrement en effectuant des requêtes différées, l'appel de service extérieur à votre application et la mise en œuvre de lien de communication entre le serveur et les client. Lors de cette application vous devrez respecter autant que possible le principe d'architecture REST (partiel REST)

Le travail rendu sera évalué selon plusieurs critères:

- Fonctionnement du résultat produit
- Avancée dans le TP
- Respect des consignes de travail
- Lisibilité du code produit
- Organisation et choix de développement
- Découpage des problématiques et utilisation du modèle MVC (Model-View-Controller)

Le programme que vous développez consiste à faire un site de gestion de plan d'entraînement sportif pour tous sports. Ces plans d'entraînement vont comprendre des exercices qui s'effectueront dans le temps ou en répétition.

2. Pages principales

Objectif : Cette section a pour objectif de créer des pages Web classiques hébergées dans le cloud de Google. Un mécanisme de délégation d'authentification devra être géré.

2.1. Page d'accueil

Créer une page d'accueil permettant de présenter votre site et automatiquement de passer à la page de recherche.

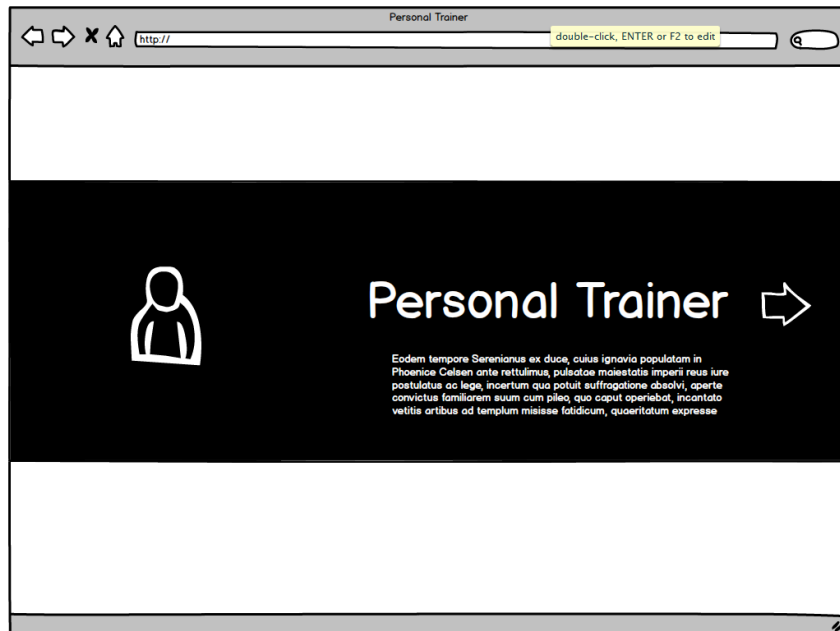


Figure 1: Welcome page

ATTENTION : le texte présent sous le titre du site sera chargé dynamiquement depuis le memory cache, si le texte n'est pas présent il faudra le rechercher dans le datastore, le placer dans le memory cache, puis l'afficher dans la page.

2.2. Page de recherche

Créer une page permettant de rechercher un plan d'entraînement ou un exercice, d'afficher les plans d'entraînement par thème et de s'authentifier.

Les fonctionnalités portées par cette page sont les suivantes :

1. Search area : permettant de rechercher un plan d'entraînement ou un exercice (**fonctionnalité traitée dans les sections suivantes**)
2. Une zone de délégation d'authentification permettant de déléguer l'authentification à google, yahoo ou openID
3. Une zone de sélection de domaine : permettant de récupérer l'ensemble des exercices et plan d'entraînement relatif à un domaine (**fonctionnalité traitée dans les sections suivantes**)
4. Lien vers la zone d'ajout de plan d'entraînement (**fonctionnalité traitée dans les sections suivantes**)

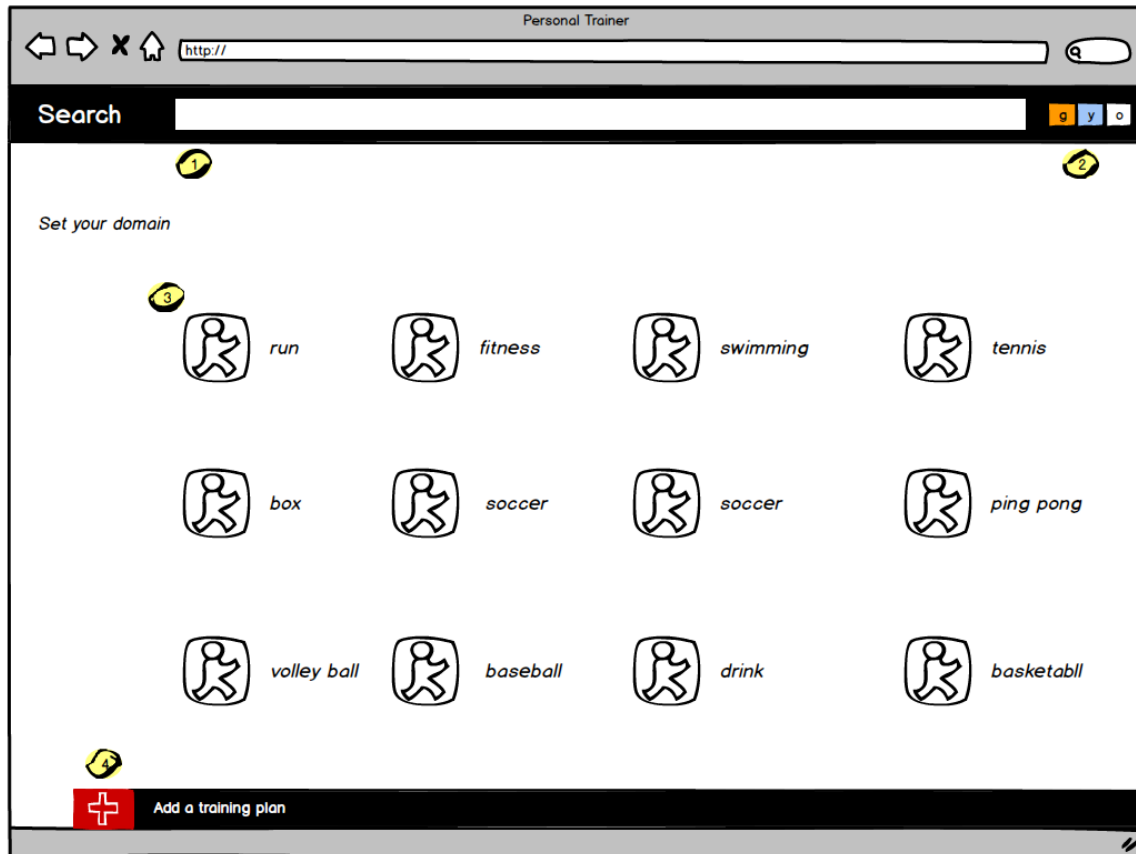


Figure 2: Search page

Réaliser la page web précédente et mettez également en œuvre la délégation d'authentification.

Une fois l'utilisateur authentifié, son nom et son email apparaîtront dans la zone 2.



Figure 3: Information d'authentification

Note : l'utilisateur ne pourra effectuer une action (recherche, ajout,...) que si il est authentifié

Cette information devra apparaître sur l'ensemble des pages de l'application. Aussi, afin de fournir cette information à l'ensemble des pages, le nom, l'email de l'utilisateur devront être stocké dans le **MEMORY CACHE**

Référence :

- <https://developers.google.com/identity/protocols/OpenIDConnect>
- <http://openid.net/connect/>
- <http://fr.slideshare.net/zeronine1/mit-2014-introduction-to-open-id-connect-and-oauth-2>
- <https://developers.google.com/identity/sign-in/web/?hl=ja>

3. Page d'ajout de plan d'entrainement

Objectif : Cette section a pour objectif d'emmener le participant à créer un formulaire d'enregistrement de plan d'entrainement et de gérer les requêtes d'enregistrement dans le Google DataStore

3.1. Page Web

Créer la page web suivante permettant de renseigner les informations d'un nouveau plan d'entrainement, les éléments suivants devront être mise en œuvre :

1. Zone de saisie respectivement title, description , domain
2. Zone d'ajout d'exercice : comprend les zones de saisie title, description, durée ou répétition. Lors de l'ajout de cet exercice, ce dernier apparaîtra dans la zone 3
3. Zone de visualisation des exercices relatifs au plan d'entrainement, en appuyant sur le bouton (3a) l'exercice sera supprimer du plan d'entrainement courant
4. Zone de temps complet du plan d'entrainement (somme des temps des exercices du plan d'entrainement courant)
5. Bouton d'ajout et d'annulation du nouveau plan d'entrainement
6. Information sur l'utilisateur (nom +email) et lien vers les données personnelles de l'utilisateur (**fonctionnalité traitée dans les sections suivantes**)

Figure 4: Add training plan page

3.2. Ajout du plan d'entrainement

3.2.1. Organisation de l'application

Votre application devra se comporter comme une application respectant les architecture REST (autant que possible), à savoir que l'ensemble des informations du formulaire seront stockés coté client (javascript) puis envoyer au serveur pour ajout en **AJAX** au format **JSON**.

Votre application serveur devra ainsi récupérer les informations et ajouter le plan d'entraînement dans le **DATASTORE**

3.2.2. Afin de permettre une plus grande fluidité d'utilisation l'ajout du plan d'entraînement devra utiliser les TASKQUEUE

4. Page de resultat de recherche

Objectif : Cette section a pour objectif d'emmener le participant effectuer des requêtes dans le datastore et a utiliser des informations provenant d'une autre source (URL fetch)

4.1. Page Web

Après avoir demandé la recherche de un ou plusieurs mots clés dans la zone de recherche, l'utilisateur sera redirigé vers l'écran suivant

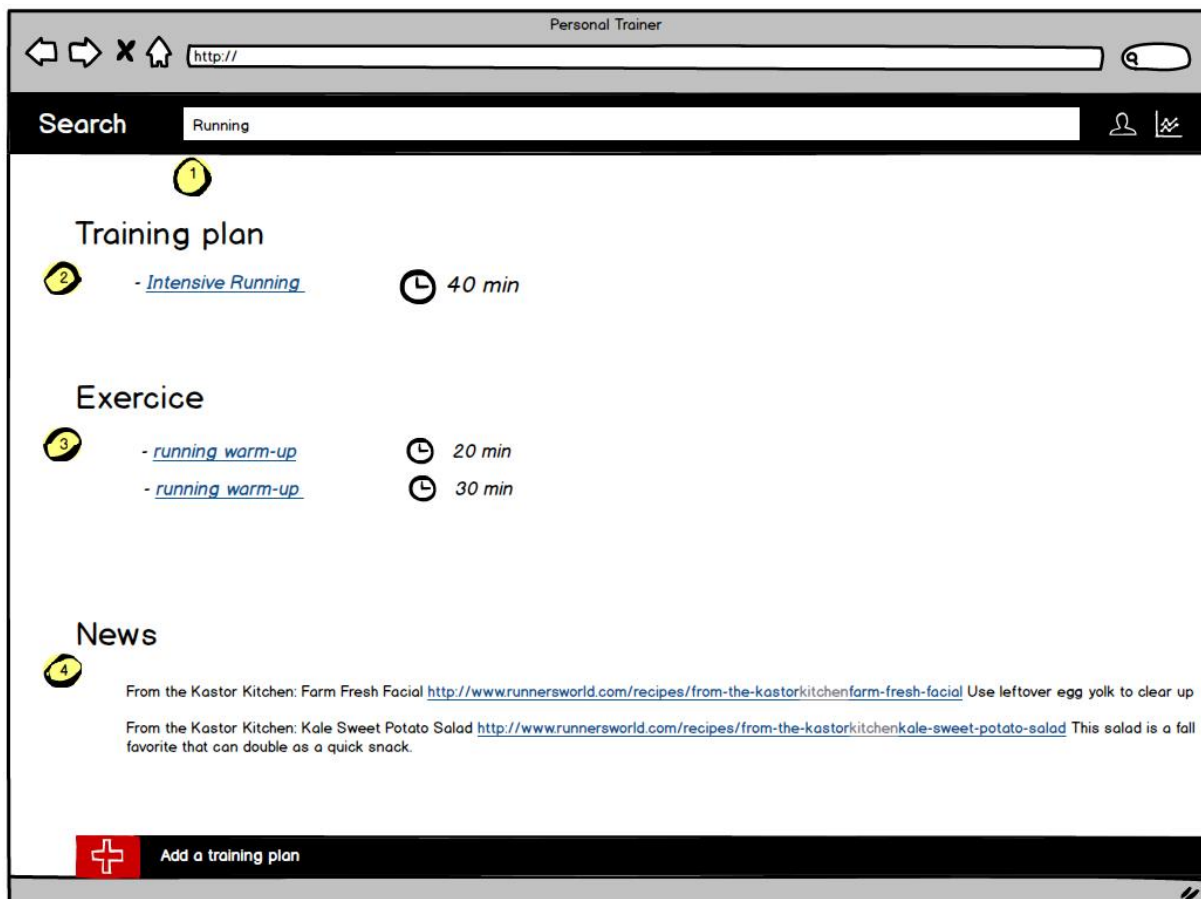


Figure 5: Search result page

4.2. Search function

La fonction de recherche devra effectuer deux recherches : une portant sur les titres des plans d'entraînement et une autre portant sur les titres des exercices. L'information extraite du DATASTORE sera retransmise en **AJAX** à la page search result puis affichée.

Les résultats de la recherche correspondent au titre (hyperlien vers description détaillée) et le temps nécessaire pour accomplir l'exercice ou le plan d'entraînement

4.3.News

A chaque recherche de l'utilisateur, des informations extraites d'un flux RSS devront être affichées dans la page. L'interrogation du flux RSS se fera à l'aide de **URL FETCH**

Exemple RSS : <http://www.runnersworld.com/taxonomy/term/740/1/feed>

5. Page de travail : détail d'un plan d'entraînement

Objectif : Durant cette partie des mises à jour régulière d'information seront envoyés au DATA STORE

5.1.Page Web

Lorsqu'un utilisateur aura cliqué sur le titre d'un plan d'entraînement ou un exercice, le détail de ce dernier s'affichera permet de démarrer et valider les exercices.

Le contenu de cette page est le suivant :

1. Titre de l'exercice à effectuer
2. Temps d'exécution de l'exercice
3. Description de l'exercice
4. Bouton de validation de l'exercice ou de passage à l'exercice suivant (note ces boutons ne seront accessibles que si les exercices précédents ont été réalisés).
5. Timer permettant de démarrer l'exercice

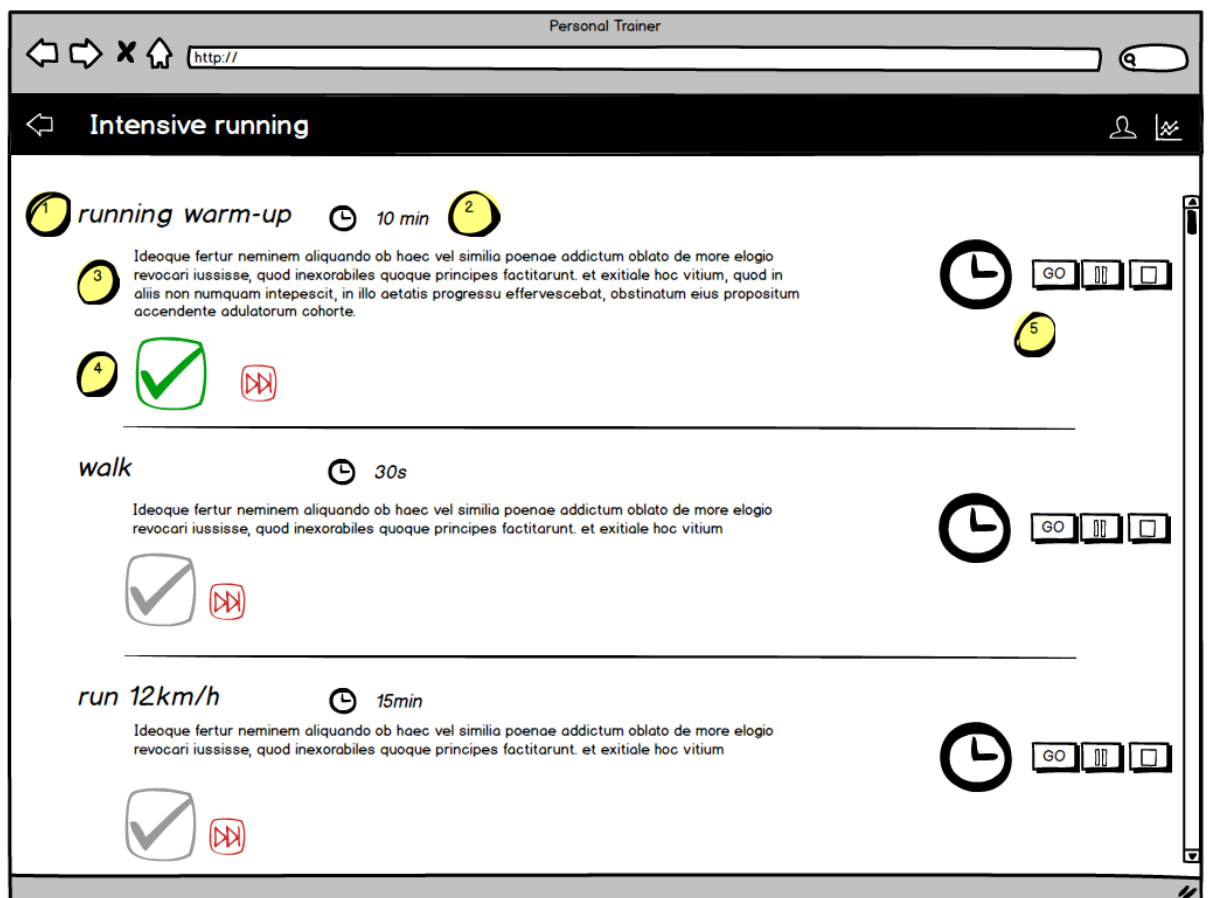


Figure 6: Detail training plan page

5.2.Comportement

A chaque validation ou passage au suivant exercice (groupe 3) une mise à jour des données personnelles de la personne sera envoyée au server les données sauvées dans le datastore seront les suivantes :

- date
- id de l'utilisateur
- titre du plan d'entraînement
- titre de l'exercice
- Success or failure

6. Visualisation des données personnels

Objectif :Requete et filtre sur le datastore

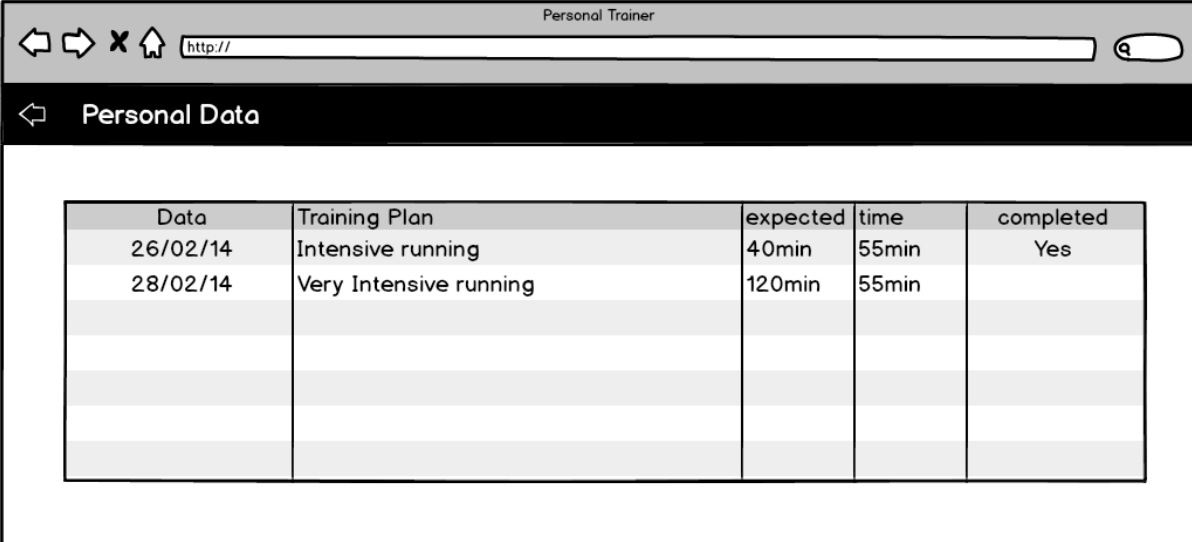
6.1.Page Web

En cliquant sur les icones dans le bandeau des données personnelles l'utilisateur pourra accéder à l'historique de ces exercices

Contenu :

- Date de l'exercice
- Type de plan d'entraînement
- Durée du plan d'entraînement
- Statut sur le plan d'entraînement

Note des informations plus détaillées comme le résultat de chaque exercice du plan d'entraînement pourront être ajoutées si le temps le permet



Data	Training Plan	expected	time	completed
26/02/14	Intensive running	40min	55min	Yes
28/02/14	Very Intensive running	120min	55min	

Figure 7: Personal information dashboard page

7. Communication avec un coach (optionnel)

Objectif : Utilisation des Channels pour des communications bi-directionnelles

7.1. Page Web

Dans la page personal data, une zone de chat est positionnée au bas de la page permettant de chatter avec les personnes connectés sur le site.

Cette zone comprend :

- Des onglets : utilisateur de chat courant, autre room de chat, ajout d'une room de chat
- Dans la zone courante de chat on retourne l'historique du chat et une zone de saisie permettant d'envoyer des messages à la personne de courante.
- L'onglet d'ajout permet de créer une nouvelle zone de chat. Elle se compose d'une zone de recherche de nom, et d'une zone d'affichage des noms trouvés (commençant par). En cliquant sur un des noms trouvés l'utilisateur ouvre un nouvel onglet de communication.

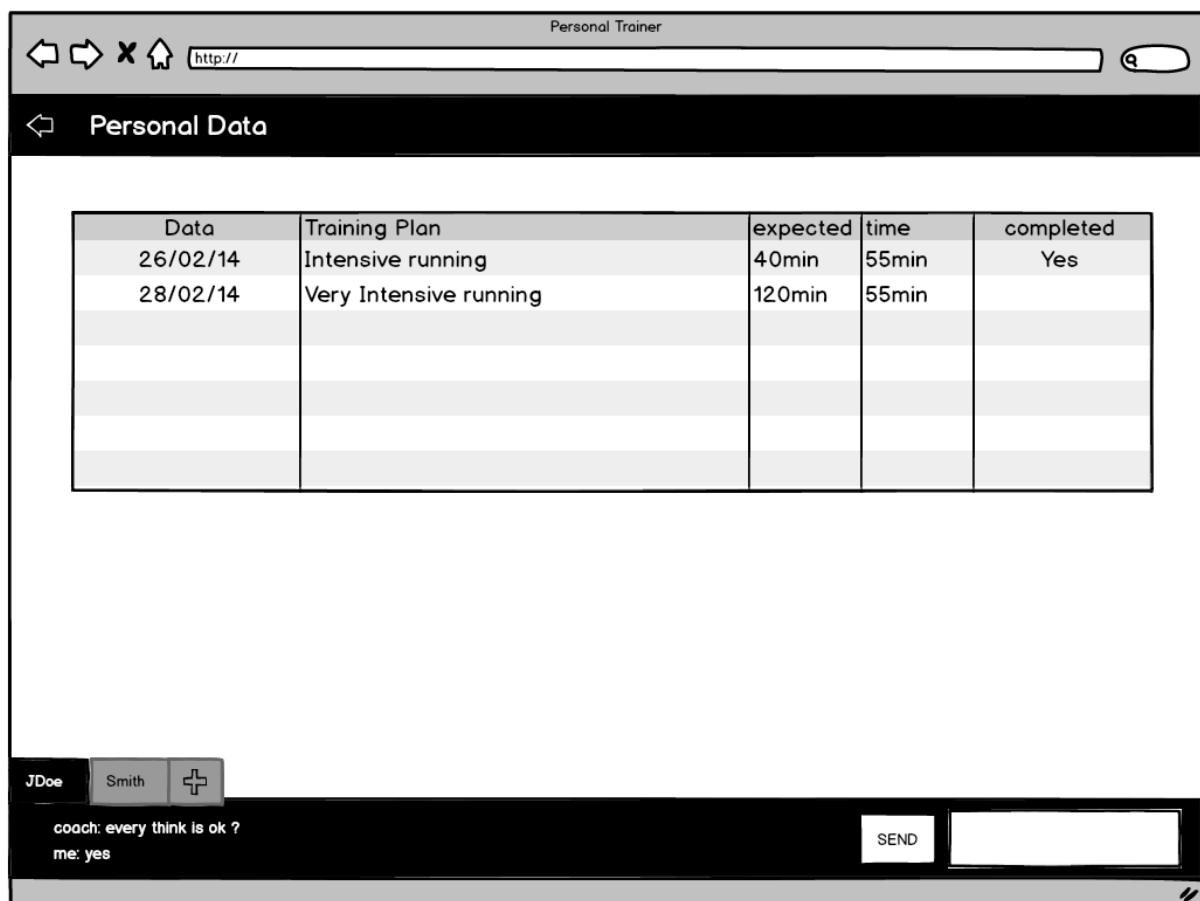


Figure 8: Chat area localisation



Figure 9: Search people for chat tab

7.2.Fonctionnement

Les utilisateurs sont enregistrés dans l'APP Cache lors de leur connection. Ainsi la recherche d'interlocuteur effectuera une recherche dans l'APP Cache

Une fois l'interlocuteur trouvé le CHANNEL API sera utilisé pour faire communiquer les interlocuteurs

L'historique des communications sera stocké dans l'APP Cache

Référence :

<https://developers.google.com/appengine/docs/java/channel/?hl=fr>