

CS GAMES 17

COMPÉTITION MOBILE

NOMBRE DE PARTICIPANTS	2
NOMBRE DE POSTES	LIBRE
VALEUR TOTALE	6%
DURÉE TOTALE	3 HEURES

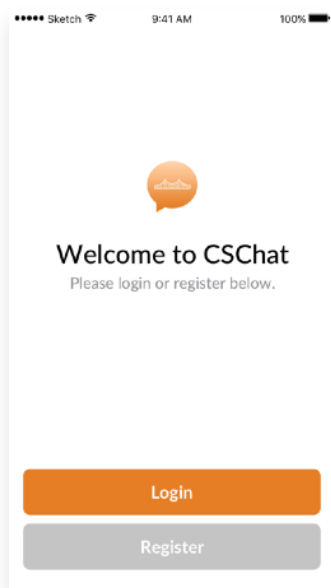
L'HISTOIRE

Samedi 25 mars 2017, 14h00. Votre entreprise technologique en démarrage se démarque déjà bien des autres : les VC observent un grand talent dans l'ensemble de votre équipe. Curieux de l'étendue de vos capacités, ils aimeraient valider un aspect primordial de votre potentiel d'épanouissement – en effet, toute bonne startup de notre époque vise à rejoindre leurs utilisateurs potentiels via la mobilité, en plein cœur de leur vie quotidienne.

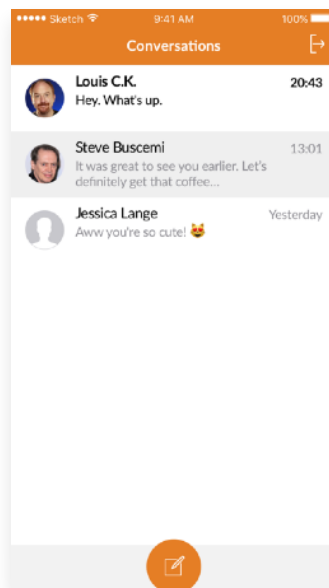
Les VC cherchent donc à évaluer votre expertise dans ce domaine en essor, sachant qu'il peut s'avérer un point majeur de succès ou d'échec de votre entreprise. Pour ce faire, ils souhaitent observer votre angle d'approche face à la plus commune des avenues: la messagerie temps réel. D'innombrables entreprises ont voulu s'inspirer de **Slack** et ses multiples homologues ; quelques-unes ont percé, beaucoup ont échoué.

À l'aide d'un projet démo déjà fonctionnel créé pour l'occasion, appelé **CSChat**, on vous demande de construire **la** tournure de l'application qui, selon vous, fournirait tout l'appât nécessaire pour devenir le prochain **Messenger** de ce monde.

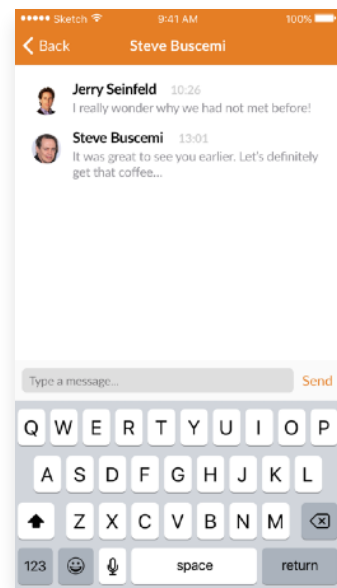
Nous connaissons bien le domaine – montrez-nous de quel bois vous vous chauffez! 🔥



Écran d'accueil



Écran de conversations



Écran de messages

LE DÉFI

Construire la nouvelle révolution de la messagerie temps réel sur téléphone intelligent.

À partir du projet **CSChat**, monter une application qui se démarque de la compétition par des fonctions innovatrices ou une approche renouvelée.

POINT DE DÉPART

Le code source de **CSChat** est disponible ici :

<https://www.github.com/mirego/csgames17-competition>

Le répertoire contient les projets de base pour *iOS* et *Android*, que vous pouvez reprendre pour débiter votre développement. Il n'est pas obligatoire de les utiliser, mais l'application doit être construite sur les SDK natifs (Cordova, Xamarin etc. ne sont pas acceptés).

Une base de serveur web a été construite en Node.js afin qu'il soit facile de le modifier. Les données sont persistées à l'aide d'un stockage intégré, donc le projet ne requiert pas de système de base de données.

Chaque projet possède sa propre documentation dans son fichier **README**, incluant toutes les instructions de configuration. Commencez d'abord par cloner l'ensemble du répertoire Github, puis suivez les directives pour compiler et démarrer l'une des applications mobiles ainsi que le serveur web.



OBJECTIFS

On ne vous demande pas de créer un clone complet de **Slack** ou **Messenger** en 3 heures, on n'est pas des fous. Cependant, nous voulons vous laisser l'opportunité de choisir *qu'est-ce* que vous allez implémenter et *comment* vous vous y prendrez.

Vous n'avez **pas** à atteindre tous les objectifs, ni toutes les fonctionnalités listées; concentrez-vous sur *ce qui vous inspire le plus*, en fonction de vos intérêts et aptitudes.

OBJECTIF #0 : FAIRE UNE APP COOL!

Selon nous, un objectif important pour votre équipe serait de nous impressionner. Lorsque vous remettrez votre travail, laissez-nous savoir qu'est-ce qui le rend cool – qu'il s'agisse d'une belle transition, d'une façon innovante de partager des messages, ou du code bien conçu.

OBJECTIF #1 : ÉCHANGER DU CONTENU

Les **Slack** et **Messenger** de ce monde n'auraient jamais percé si leurs utilisateurs ne pouvaient pas communiquer en temps réel. Nous vous avons fourni une application mobile de base et un serveur web qui permettent une communication simple. Ajoutez-y ces fonctionnalités :

- Permettre des **échanges en temps réel** de contenu texte entre les utilisateurs
- Permettre des **échanges en temps réel** de contenu multimédia entre les utilisateurs
- Permettre à l'expéditeur de **créer de nouvelles conversations** avec un ou plusieurs utilisateurs, puis d'être notifiés lors des nouvelles communications
- Permettre aux messages d'être **privés**, et donc accessibles seulement entre les interlocuteurs concernés
- Permettre aux utilisateurs de **s'inscrire** et de **se connecter sécuritairement**

Tel qu'expliqué, vous n'avez pas à faire tout ça – choisissez ce qui vous intéresse le plus.

OBJECTIF #2 : CONSTRUIRE DES MESSAGES

Échanger du contenu, c'est amusant, mais nous pouvons rendre le tout plus personnel en permettant aux utilisateurs de personnaliser leurs messages :

- Permettre d'envoyer du **texte riche** (Markdown, formatage WYSIWYG)
- Permettre d'envoyer des **contenus spéciaux** (stickers, GIFs, animations plein-écran)
- Permettre d'initier des **appels vidéo** (de façon similaire à FaceTime ou Skype)
- Permettre de **styler l'application** (changer le thème, les couleurs, la police)

Encore ici, c'est seulement de l'inspiration – ajoutez ce qui vous semble le mieux!

SOUSSION

Gardez 10 minutes à la fin de la compétition pour préparer votre soumission du projet.

Ouvrez le fichier **SOLUTION.md** dans le répertoire du projet, puis remplacez le texte entouré de soulignés. Cela nous permettra de voir ce que vous *planifiez* faire et ce que vous avez *réussi* à faire. Prenez le temps de bien expliquer chaque détail de votre solution – nous ne voudrions pas en manquer des bouts.

Ensuite, supprimez les dossiers suivants du répertoire du projet :

- ✓ **android/build**
- ✓ **ios/Pods**
- ✓ **server/node_modules**

Cela permettra de réduire la taille du projet et évitera de congestionner le serveur.

Une fois votre solution bien décrite et votre projet nettoyé, archivez le dossier dans un fichier zip, puis soumettez-le à l'aide du système de remise fourni par l'organisation.

POINTAGE

Nous allons évaluer votre travail en fonction de **comment vous avez fait** chacune des fonctionnalités, et non le **nombre de fonctionnalités** que vous avez essayé d'implémenter. Il est mieux de faire *une seule chose très bien* que de faire *plusieurs choses incomplètes*.

Vos solutions seront notées à l'aide de la grille suivante :

Critère	Points
Solution <ul style="list-style-type: none"> - Résolution générale du problème - Originalité de la solution - Qualité de l'intégration (apparence et utilisation) - Respect du thème 	/ 50
Implémentation <ul style="list-style-type: none"> - Qualité générale du code - Bonne utilisation des <i>patterns</i> de programmation - Bonne utilisation des librairies et ressources externes - Respect des standards de la plateforme 	/ 30
Qualité <ul style="list-style-type: none"> - Respect de la structure du projet - Respect de la syntaxe du langage 	/ 10
Soumission <ul style="list-style-type: none"> - Description de la solution fournie et bien documentée - Archive du projet nettoyée et facile à utiliser 	/ 10
Total	/ 100

« ONE MORE THING »

Concentrez-vous sur vos forces et impressionnez-nous.