



MOBILE COMPETITION

PARTICIPANTS

2

WORKSTATIONS

UNRESTRICTED

VALUE

5%

DURATION

3 HOURS

THE STORY

During the pre-Alpha days, **Snapchat** was one of the top communication companies in the world. Entirely mobile-based, it survived the disaster as the only remaining messaging service, a modern equivalent of the *Pony Express*. It gained a place as part of the Empire as the **key communication backbone**, over which they never ended up taking control. The company may analyze every message, and could keep track of *everything* happening inside the Dome.

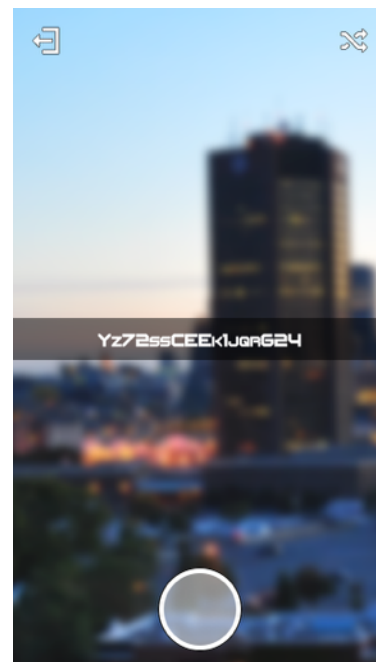
For the people of the Empire, there's a growing concern over **privacy**. They feel the need for their own **private channel**, to communicate with each other without an external oversight.

As proclaimed technical experts, your legion was summoned by the Empire to build the **prototype for a new communication service**. They want to be impressed, as the best team will be invited inside the Dome to complete the project. *But they're not giving you a lot of time.*

Inside the Rebellion, an old legion once built a *Snapchat* clone, called **Rebel Chat**. It was quite barebones – it could only send a random string of characters to a local server. But you have access to the source code, so you can use it as a base to **create something awesome**.



Rebel Chat home screen



Rebel Chat message screen

THE CHALLENGE

Build a **Snapshot** clone, with one key difference: instead of sharing *pictures* and *videos* taken with the camera, share only *texts* and *drawings*. (Because, really, who wants to share images of the world as it is now?)

STARTING POINT

The full source code of **Rebel Chat** is available here:

<https://www.github.com/mirego/csgames16-competition>

It includes the base projects for *iOS* and *Android*, so you can choose the platform on which you want to develop.

The web server is built in Node.js to make it easy for everyone to extend. The data is persisted using an embedded data store, so it doesn't require any database system.

Each project has its own documentation inside the **README** file, including setup instructions to get started. First start by cloning the full repository from Github, then follow the directions to build and run the mobile applications and the web server.



GOALS

We're not asking you to build a full **Snapshot** clone in 3 hours, we're not crazy. However, we want to give you the opportunity to choose **what** you will implement and **how** you will do it.

You **don't** have to meet all the goals, or every feature listed in each; focus on **what inspires you the most**, based on your interests and abilities.

GOAL #0: MAKE SOMETHING AWESOME!

We think that an important goal for your team should be to impress us. So, when you hand-in your work, let us know what you think makes it awesome – be it a cool UI transition, an innovative way to share data between two users, or awesomely crafted code.

GOAL #1: SHARING CONTENT

Snapshot would have never made it if users could not *share* with each other. We've provided you with a sample mobile application and web server that allows very basic communication.

Expand its capability by:

- Allowing **exchanges** of text messages between users
- Allowing **exchanges** of image messages between users
- Allowing the sender to **choose recipients** for his message, and be able to receive messages from the receiver part
- Allowing messages to be **private** and only shared between the sender and the intended recipients
- Allowing users to **register** and fully *sign-in* into the app

As explained above, you don't have to do all this – select what you think matters the most.

GOAL #2: CRAFTING MESSAGES

Sharing content is fun, but we can make it more personal by having users customize the content and the visual aspect of their messages. Introduce customization in the app by:

- Allowing the user to **draw** on the screen (with multiple colors / types of strokes)
- Allowing the user to **erase** parts of the drawing
- Allowing the user to edit the **text caption** (move the label, change its appearance, select another font)
- Allowing the user to change the **background picture**

Again, this is just for inspiration – implement what you think would make the app better.

SUBMISSION

Keep 5 minutes at the end of the competition to prepare your project for submission.

Open the file named **SOLUTION.md** in the project directory, and replace the texts surrounded by **underscores**. This will help us see what you *planned* to do and what you *ended up doing*. Take your time to explain every detail of your solution – we don't want to miss any part of it.

Then, delete the following folders from the project directory:

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

This will help reduce the size of the project and avoid overloading the server.

After your solution is correctly described and the folders are clean, archive the entire project directory in a single zip file and submit it through the system provided by the organization.

SCORING

We will score your work based on **how well you did** on each feature, not the **number of features** you did try to implement. It's better to do *one thing really well* than do *multiple things incorrectly or incompletely*.

Your solutions will be marked using the following grid:

Criteria	Points
Solution <ul style="list-style-type: none"> - General resolution of the problem - Originality of the solution - Quality of the integration (look and feel) - Respect of the theme 	/ 50
Implementation <ul style="list-style-type: none"> - General quality of the code - Good use of programming patterns - Good use of libraries and external resources - Respect of the platform standards 	/ 30
Quality <ul style="list-style-type: none"> - Respect of the project structure - Respect of the language syntax 	/ 10
Submission <ul style="list-style-type: none"> - Solution description provided and well-documented - Project archive cleaned and easy to use 	/ 10
Total	/ 100

ONE MORE THING

Focus on your strengths and impress everyone.