



## Project: Rocking Socket.

### Prologue:

If we are to say about the project in two words, we'd say "Socket ". See, we barely needed two words.

### En Détail:

We need a build a **local** service for our **inter team** communication. And We need to build something that can **send** and **receive** messages in real time. So to meet the "**real time**" scenario, we might need to adapt cooler technology like **socket**. This service can be usable by any end user. Meaning we need to build an **UI** that actually works. Through that UI, we will send texts to each other. We need to be able to receive others' texts too.

See, as we are adding more sugar to this service, we are making it heavy, but we improvise you to build it. It's gonna be, ( wait for it), **legendary**.

### What You Have To Accomplish:

There are several goals to this accomplishment. We better enlist it as follows.

#### Milestone Uno:

You have to answer some questions.

1. Which **protocol** are you gonna use? UDP or TCP or somethin' else, why?
2. What is **your plan** on this project to go through?
3. What are the **technical challenges**( Ex. Scaling it via Multithreading, etc ) to achieve the desired outputs?
4. What is **your thought** on this socket thing support in **Java**? How do you cope with it?
5. Have you ever built an **End-to-End** chat application like this one? If yes, was it scalable enough?

#### Milestone Dos:

Bootstrapping your project.

1. Set Up Your **project** and write a **Client- Server** demo application.
2. Open a **server** on a certain Port.
3. Let the clients join.
4. Bound a list of **rules** on that port. ( Ex. certain clients can join).

5. Find out how many **clients** does the server can serve up to.

### **Milestone Finale:**

Let's Build a **Simple UI** for our Application.

1. Define how do you provide a **GUI** for your User.
2. Build a Super Simple GUI for this project.
3. Wrap it over your **Java CLI** application.
4. Prepare a Single Super Simple **Documentation** of this project. It should contain how to run, what to install and how to operate.
5. Pat On your shoulder and Have a cup of **Tea**.

### **Where are some"Hints:**

1. If you are total **newbie** on Socket, please at first Go through some good reads of your choice.
2. For better **IDE** you are open to use **IntelliJ** or **Eclipse**.
3. We suggest for **GUI**, you stick with **Java Swing** or **JavaFx**. By sticking with "**All-Java**" ecosystem it would be easier to build such application.
4. Start from the **Milestone Uno**. And then develop gradually. By completing along, you will finish the project just by time.
5. We want to **follow** along with you on how you are progressing via **Github**. So it is mandatory, that you mail us the link as soon as you start the project as **rocking\_socket** and put it on the github repo.

### **Cliche:**

That's it. Feel free to follow your gut and **intuitions**. You are not bound to follow the "**Hints**" we provided. Amaze Us. :)