Final Project – Ticket Sale Application
June, 2018

1. how to run the project:
   1.1. Open 2 terminals and go to the file directory where it contains .sbt file.
   1.2. Enter ‘sbt’ in both terminal, and then enter ‘run’ in both terminal as well.
   1.3. In one terminal enter ‘1’, and another terminal enter ‘2’.
   1.4. See results.

2. Overall system design
   2.1. This is a system for selling tickets, I created a set of ‘kiosk’ actors those receive chunks of ticket for each event from a master actor. Clients connects one of the kiosks to purchase tickets. When a kiosk runs out of tickets, it requests another chunk of ticket from the master. If the master has no more chunks and other kiosks has no more left either, the ticket is sold out. If the master has no more chunks but another kiosk does, it transfers tickets to the one needed. I did so by creating a ring to synchronize the actions between master and kiosks.

3. Design decisions
   3.1. I decided to separate the project into 2 different folders: client and server. It makes the whole design process clear.
   3.2. Client actor has three status which displays different messages.
   3.3. Master is in charge of sending chunk of ticket and insures no failure.
   3.4. In Kiosk for case *Msg*, I used three variables to check the tickets status. To find out where are they, are they sold out etc. Other cases are the main part of this application.
   3.5. Tried to make this application distributed but fails to do so.

4. Synchronization and Fault tolerance aspects
   4.1. Synchronization: an important part in this application, actions are synchronized between master and kiosks.
   4.2. I used supervisor strategies to define fault tolerant behavior of the system. I used OneforOneStrategy to resume, restart, stop the children and stop supervisor itself.