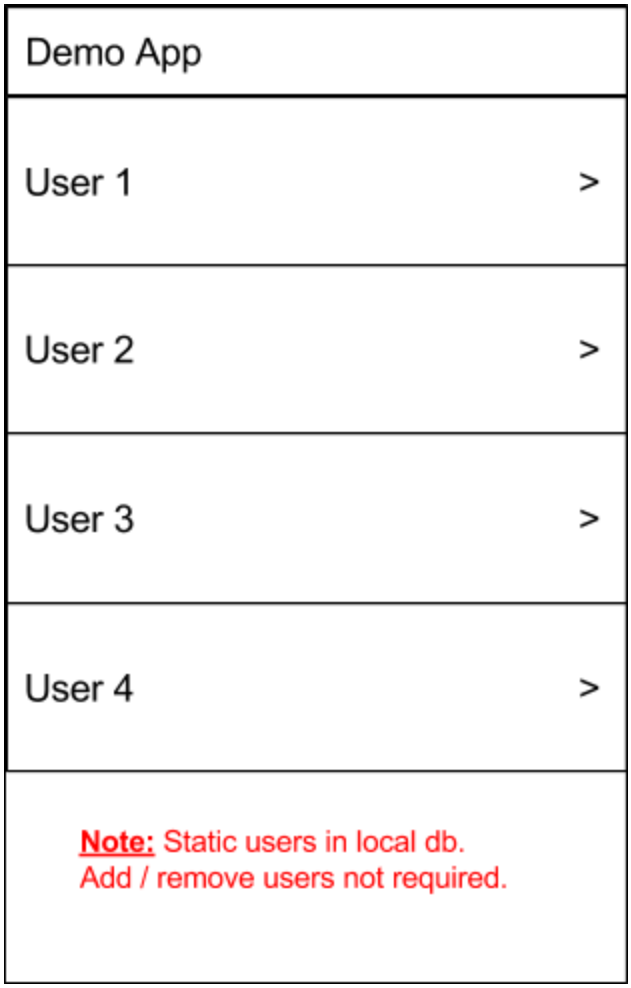
# Akonect - iOS Developer Task

Create a simple native iOS app using ObjC that connects to Socket.IO server and performs the following tasks.

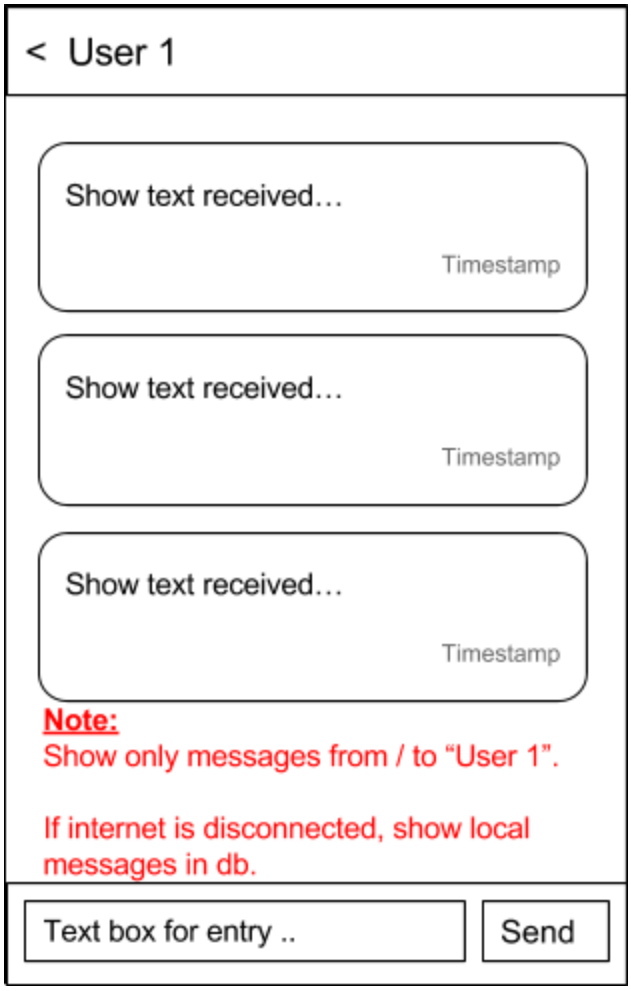
Reference:
http://socket.io

## App Wireframe

Screen 1



Screen 2




## 1. Connect to socket service

Connect to the socket server and listen to events.

**Socket endpoint:** http://54.89.42.119:3009
**Socket JS (to test on web):** http://54.89.42.119:3009/socket.io/socket.io.js

---

## 2. Send a message via socket

On button “Send”, pass the text field value to the socket function “send_msg” and get acknowledgement. If acknowledgement fails, show error.

**Function (emit):**
send_msg

**Parameters (to send):**
{
txt: "sample text",
user_id: “user1”
}

**Callback (acknowledgement):**
{
status: 1 (1=success, 99 / -1 / 0 = failed)
}

---

## 3. Receive and display message via socket

Once socket is connected (and app is open), listen to event “rcv_msg”. The event will be triggered on new message on server. Once message is received, display in the app (as per wireframe).

**Function (listen on):**
rcv_msg

**Parameters (received):**
{
txt: "sample text",
timestamp: 1472126199,
user_id: “user1”
}

---

## 4. Save messages in local DB and show in correct screen

Store all received messages in local db with correct **user_id** reference. In screen 2, load messages from db as per user clicked from screen 1.

---