**Design Proposal #1**

**On Device:**

* Lightweight OS
* Preferably Custom VNC Viewer with custom protocol else we get the most lightweight one possible.
* Some sort of device startup code that immediately attempts to connect the device to the server, and shows a login page, where the student signs in and is connected to a device.
* On lack of internet connection, admin (teacher) can log into OS on the device, and reset internet connection.
  + Once connection is reset, student is shown sign in page.

**On Server:**

* Server hosting sign in page, which then forwards (proxy) students to an App running on the device (Android OS) that **they preferably cannot close.**
* The app grants access to other apps on the device (i.e. shows a fixed list of apps the student(s)) can use. This means implementation of access permissions.
* Teacher has admin access in the app:
  + They can see list of students connected.
  + Apps that students are using.
  + View student(s) screens. (When student has a question, a student would raise their hand and teacher would have to walk to them. But here they can just view the student’s screen)
  + Kick students off.
  + Remove or grant access to other apps.
  + Add of remove students from the system.
  + Share/takeover student(s) screens.

**Software:**

* Custom VNCViewer (with custom protocol) on raspberry pi.
* Docker/android emulator configuration.
* Building an android application.
* Building a server to host login.
* Server that hosts database.

**Linux Approach for Server:**

* Build a UI using Electron for teacher.
* Implement a server in each student VM that gets updates from teacher’s app.
  + The server should be capable of showing adding/removing app permissions for the student.
  + Providing teacher client with information of apps active/inactive per students computer.