RFID keycard authentication

In addition to the interface for staff users, the system also authenticates keycards and sends other information to the microcontroller (Arduino) that connects to the RFID scanner and operates the locking mechanism.

Simulating authenticating individual keycards

Keycards are authenticated like this:

- Lock user scans keycard at the RFID reader.
- The Arduino sends a request to the server hosting the lock management application.
    - The request URL identifies the door and the RFID, e.g. when a user scans a keycard with the RFID 9999999999 at a door with the id 1, the URL contains "/checkdoor/1/checkrfid/9999999999".
- If the response content is "1," the Arduino unlocks the door.

Since the lock management application is being developed independently of the lock hardware involved (not counting collaboration concerning communication for authentication, e.g. request URL syntax and response content format), we need a way to simulate keycard scanning without involving the Arduino.

So, to simulate the process, simply go to http://xx.xx.x.x:yyyy/checkdoor/<door_id>/checkrfid/<rfid_id>/ on the same network, where xx.xx.x.x is the ip of the computer running the development server; yyyy is the port; <door_id> is an integer denoting a particular door; and <rfid_id> is a 10-character string representing the RFID. For example, if you are running the development server with the command

```
$ ./manage.py runserver 11.11.1.242:9999
```

you can go to http://11.11.1.242:9999/checkdoor/2/checkrfid/ABCDE12345/ from any device on the network to simulate a keycard (with RFID ABCDE12345) scan at door with the id 2.

(Note: this process is the same as simulating scanning a keycard for the purposes of assigning it, as described in the section "Adding a new lock user and assigning a keycard" above.)

Simulating retrieving *list* of permitted RFIDs

The Arduino periodically caches lists of authenticated users so that the lock can still function in case it can't communicate with the Django application. Obtaining a list of authenticated users for a particular space is similar to the single keycard authentication process described above:

- The Arduino sends a request to the server hosting the lock management application.
    - The request URL identifies the door, e.g. when a user scans a keycard with the RFID 9999999999 at a door with the id 1, the URL contains "/door/1/getallowed/".
- The response contains the RFIDs allowed to access that particular space, in JSON format, e.g
  `{"doorid": 1, "allowed_rfids": ["1122135122", "9999999922"]}`

To simulate this, go to http://xx.xx.x.x:yyyy/checkdoor/<door_id>/getallowed/ on the same network, where xx.xx.x.x is the ip of the computer running the development server; yyyy is the port; and <door_id> is an integer denoting a particular door. For example, if you are running the development server with the command

```
$ ./manage.py runserver 11.11.1.242:9999
```

you can go to http://11.11.1.242:9999/door/2/getallowed/ from any device on the network to get all RFIDs allowed access to this door.