

Tag Me Home

A social services queueing system

Vijeta Karani, Stephen Misson, Ashish Pai, Raina Shastri

Computer Engineering
San Jose State University
San Jose, CA, US

Abstract—Tag Me Home is a queuing and bed reservation management system, designed to provide fast and easy bed reservations for homeless shelters. It consists of a web application for homeless shelter administrators and a collection of applications for NFC (near field communication) tag readers.

I. INTRODUCTION

In big cities around the world, there are people who have lost their homes because of a string of bad luck, or other circumstances that can't be helped. To keep these unfortunate people from having to sleep on the street or scrounge for a meal, often big cities provide services for the public welfare. Perhaps a shelter will provide a place to sleep, a hot meal, job search services, or many other services. These services are very helpful, but they are often inefficient. For example, homeless people have to spend a large amount of time waiting in long lines at various services around the city. Our product, Tag Me Home, aims to combat this particular inefficiency using NFC (near field communication) tag technology, a cloud database, and a web administration portal. The NFC tag technology is a fast and simple way of providing identification which can be used to check-in to social services. The cloud database ensures that the system is easily deployed and easily accessible by all other social services center. The web administration portal provides a familiar administrative user interface for managing resources. Our team provides a complete system to manage beds at a homeless shelter using these components. This system could be extended to other social services that require check-in.

II. NFC TAG READER APPLICATIONS

The NFC tag reader applications provide all functions which require NFC tag reader hardware. Three applications are provided.

The first application registers a new NFC tag. Before starting the application, the user's information should be entered as a user record, including a unique number, called the "tagNo". Then, an administrator starts the application and enters the "tagNo". An NFC tag needs to be placed on the tag reader, so it can be programmed. The tag is programmed with an encrypted copy of an automatically generated unique number, the "rfidCode". This number and the "tagNo" are

stored as a record in the database. Then the tag registration is complete. This whole process only takes a few seconds.

The second application checks in a user to a bed. The user must first scan the NFC tag. Then the system searches the database for an available bed. If a bed is available, the user record is updated with the bed id, and the bed record is updated as occupied. If a bed is not available, it provides this information to the user.

The third application checks out a user from a bed. The user scans the tag and the program searches the user record for the bed id. It then removes the bed id, and updates the bed record to available.

These three applications should be used at three different places in the facility. The check-out application at the exit door, the check-in application at the entrance, and the new user application can be anywhere convenient for the administrator.

We used Python to write these applications, because there were available libraries to interface with NFC hardware (nfcpy [1]), with MongoDB (pymongo [2]), and a cryptography library (pycrypto [3]) to ensure identification was encrypted on the NFC tag. It was also a familiar programming language for the engineer assigned to the NFC tag reader applications. Libusb [4] was used as a driver for the tag reader. The tag reader used was manufactured by Advanced Card Systems Ltd., model ACR122.

III. WEB ADMINISTRATION PORTAL APPLICATION

The web administration portal is a cloud application and uses MongoLab [5] - a cloud database, for storing the demographic details of homeless people and keeping a track of the bed availability. This cloud application offers a number of administrative operations that help the staff at the social services, in keeping a track of the users registered for the bed and in viewing the bed status. The administrator at the social service centers can mainly perform the following operations through the web administration portal –

- a. Register the homeless person at the social service center.
- b. Search for a particular person's demographic details.

- c. View all homeless people at the social service center for registered for availing the bed service. Also displays the RFID tag and the bed number assigned to each person.
- d. Update the demographic details of the registered person. Also, allows assigning a bed number manually.
- e. Delete the registered person.
- f. View the beds existing at the social services center, along with their status – available or occupied.
- g. Delete the beds existing at the social services center.

We used Java – REST services [6] to offer the above mentioned services for the web administration portal. REST API in java offers higher level of flexibility in building dynamic operations and requires minimum amount of changes in the event of new operations being added to the web administration portal. The user Interface has been created using HTML5 and CSS3. The The User Interface has been connected to the JAVA REST services using jQuery, AJAX and JavaScript.[7]

The web administration portal application has been deployed to AWS [Amazon Web Services] as a cloud application using Amazon EC2 for Ubuntu server.[8]

IV. CONCLUSION

Therefore, to conclude, our aim is to reduce the waiting time spent by the homeless people waiting at the social services centre for availing the bed service. We have achieved this through the NFC technology. Instead they can utilize the time to find job and improve their life and come out of the homeless state. Through the web administration cloud portal, we aim to provide one single view of the homeless people registered at the social services centre and provide an efficient way of tracking the user and the respective bed details.

- [1] S. Tiedemann. (2014, December 8). Nfcpy.org [Online]. Available: <http://nfcpy.org>
- [2] B. Hackett. (2014, December 8). mongodb/mongo-python-driver [Online]. Available: <https://github.com/mongodb/mongo-python-driver>
- [3] D. Litzemberger. (2014, December 8). PyCrypto - The Python Cryptography Toolkit [Online]. Available: <https://www.dlitz.net/software/pycrypto/>
- [4] D. Drake, J. Erdfelt, N. Hjelm, P. Batard, M. Plante, H. de Geode, et al.. (2014, December 8). libusb [Online]. Available: <http://www.libusb.org/>
- [5] Available: <https://mongolab.com/>
- [6] Available: <http://docs.oracle.com/>, http://en.wikipedia.org/wiki/Java_API_for_RESTful_Web_Services
- [7] Available: <http://www.w3schools.com/js/> , <http://api.jquery.com/> , <https://developer.mozilla.org/enUS/docs/Web/Guide/HTML/HTML5>
- [8] Available: <http://aws.amazon.com>