

Spacehold

Nancy Ouyang, Sherry Wu, Victor Youk

May 13, 2012

1 Abstract

Members of shared group spaces who do not have key access often are unaware that the space is open. To remedy this lack of information, an Android application was developed. It allows for automatic check-ins when a keyholder enters the space and deregisters them when they leave. With the use of this application, members at the MIT Electronics Research Society reported that they were more likely to visit the space because they could easily check whether the space was open.

2 Introduction

Makerspaces, hackerspaces and other places that promote access to tools are a growing phenomenon around the world, garnering the attention of even the White House which invited a group of students from Georgia Tech’s Invention Studio to visit this past week. The relative novelty of these spaces means that many are encountering common problems that could be solved. This project focuses on addressing one of the needs of the student-run shop on MIT’s campus, MIT Electronics Research Society (MITERS). Specifically, members expressed an interest in better broadcasting the state of the shop to the general MIT community to encourage students to use the shop for projects. Currently, a set of authorized users, “keyholders,” are given the code to access the shop 24/7. As long as a keyholder is there, the shop may be used by anyone and it is the keyholder’s responsibility to ensure safe practices on tools such as high voltage power

supplies, the mill, band, or the lathe [Figure].

3 Related Work

(look for keyholding clubs – IIRC there wasn’t many and they all sucked)

The difference between Spacehold and competitors is that Spacehold presents a nonintrusive and fully automatic system. (elaborate)

4 Background

The primary motivation of this project is to build an automatic keyholding system for the on-campus and oft-visited makerspace, MITERS, the MIT electronics research society. The club’s existing keyholding system, which entails logging onto a website, checking one’s name, and entering the projected duration of one’s keyholding time, has been sufficiently out of the way to warrant keyholders not signing in, voiced by many members. In addition to requiring multiple steps to broadcast one’s presence, it involves direct interaction with a computer, which people might not bring to MITERS since it is mainly a hardware hackerspace. MITERS offers a public computer for those who do not bring laptops, but it is often in a nonfunctioning state for registering one’s presence. As a result, the current system is not compelling for keyholders to broadcast their presence.

In addition to the two biggest complaints, dwellers at MITERS wanted easy sharing features for documenting projects on the web. Many members currently have blogs, which they

say “could be updated way more frequently because I have to take out my camera, take a picture, upload it to my computer, and upload it to my blog through a clunky interface.”

Many interviewees in the field test also said this idea could be broadened to more than just makerspaces. For example, members of a club with a restricted-access space, which may contain valuables instead of dangerous machinery, could use the app. Living groups could leverage the location-aware alerts to see who is at the house at a given time.

5 System Description

The Spacehold system is designed from the ground up to be simple for users and administrators. It consists of two parts: the backend and the frontend. The backend will be discussed first to establish a foundation with which the frontend will communicate. The frontend consists of the Android application, the main project, and a web application for the club president and any other administrators to customize the service.

The backend consists of a centralized server run by the Spacehold developers (heretoforth known as Spaceholders) to eliminate club spaces from acquiring and setting up their own infrastructure, thus reducing maintenance and simplifying the system for the consumer. It will store data for all clubs, such as introductory information about the club, the location of the club, a list of authenticated keyholders as dictated by the club president, and the list of keyholders and club members currently at the club. The backend will mark the club as open if there is at least one keyholder at the space as deemed by the server, and closed if there are no keyholders at the space according to the server. Initially, servers from the Student Information Processing Board (SIPB) will be used because they are reliable and can handle the load required by Spacehold.

The Android application is the heart of the project. It has one view containing the state

of the club and the list of keyholders if the club space is open. When a keyholder with Spacehold installed on her phone is within 50ft of the space as determined by cellular and WiFi triangulation for 15 or more minutes, Spacehold phones home to the backend to broadcast that this user is keyholding at the space. However, if a club member is within 50ft of the space for at least 15 minutes, then Spacehold phones home and add him to the list of club members currently at the club if there is at least one person keyholding. When a keyholder or club member is away from the area for 15 minutes or more, then Spacehold tells the server to remove him from the list of members currently at the space [cite].

Excellent WiFi triangulation accuracy was achieved using a priori knowledge of the locations of wireless base stations at a location, in this example, Massachusetts Institute of Technology. A list of base station BSSIDs and their respective locations was obtained from IS&T. When the phone connects to a new base station, Spacehold checks the location of the station and determines whether the user is at the club.

Although the aforementioned a priori knowledge of base station locations tremendously increases the accuracy of Spacehold, there are occasions when this fails. For example, MITERS is located immediately adjacent to the Tech Model Railroad Club. To reduce the number of incorrect keyholding alerts, when the app thinks the user is at the club, it will vibrate the phone and ask the user to reject the impending keyholding if the user is at a neighboring club. After 15 seconds (on top of the 15 minutes for avoiding passing-by errors), the message will disappear and the user is reported as currently at the club.

To minimize setup time and other hassles, the OAuth API was leveraged for easy authentication with one’s Google account, which virtually every Android user uses with their phone. This reduces the need to create and maintain a custom registration for the developers and yet another set of username and password for the user

to remember.

(write about Media API used)

A secondary front end, a web application, was developed. It is used for controlling server preferences by makerspace administrators and as a way for people without the application to view whether the space is open and, if it is, who is currently keyholding. In addition, it can serve as a backup, manual keyholding registration if a keyholder's phone is out of service. Similar to the mobile application, the web application keyholding entry will attempt to deduce the user's location using the computer's IP address, which Google does to give more location-relevant search results.

6 Field Study

Your mom

7 Discussion

The client/server protocol could be greatly improved. The most important vulnerability as of today is that anyone can communicate with the server if they know the protocol and the authentication key and listen to the data transmission since it is all plaintext. In an ideal world, the authentication token would be on the order of twenty characters long, similar to other services keys, e.g. Remember the Milk, and communication would be encrypted and transferred over SSL.

Instead of using PHP get requests to transfer inline arguments, the app and server could communicate with more robust and more programming-friendly REST requests, which transmit JSON. Although this will incur a greater overhead, it is able to describe more types of objects the application may use in the future, such as keyholding messages.

A third usability note is that the user interface of the application can be improved. For the purposes of a prototype, most of the work was focused on creating a functional application

as detailed in the proposal. Now that the core functionality is in the application, more time and effort can be spent cleaning up the user interface to make using the application more intuitive and enjoyable.

8 Conclusion/Future Work

Your mom

9 References

Your mom