Utilizing Multiple Location-Based Service APIs for an All-In-One Mobile Application Check-In Solution

David Ivins

Drexel University  
3141 Chestnut Street  
Philadelphia, PA 19104 USA  
+1 856 371 0204  
davidivins@gmail.com

# ABSTRACT

This will be the abstract section

## Keywords

This, is, the, keywords, section

# INTRODUCTION

*LEAVE BLANK THE LAST 2.5 cm (1”) OF THE LEFT COLUMN ON THE FIRST PAGE FOR THE COPYRIGHT NOTICE.*

Ever since Apple introduced the iPhone in 2007, smartphone technology has been a dominating force in popular tech culture. Prior to Apple’s entrance into the market, businessmen and technology enthusiasts primarily used smartphones. Now, those ranging from high school students to senior citizens are carrying smartphones. In addition to increase in popularity, there has also been a dramatic increase in the amount of functionality offered by these devices. Many years ago, smartphones offered only basic functionality, such as web, email, and document management. Now they are coming equipped with global positioning systems, accelerometers, and gyroscopes as standard features. Two of the more popular operating systems included on phones of this nature are Apple’s iOS and Google’s Android. Both of these operating systems offer a software developer kit and a vast amount of documentation. Given the new phone hardware and the accessibility of the SDKs, a wealth of new and interesting mobile applications are being developed for these platforms [1]. These interesting applications cover areas such as location-based services, augmented reality, and interactive gaming [2]. Location-based services, in particular provide a number of useful features to users, including social networking, mapping, organizing, planning, searching, and sharing of information. Furthermore, through public APIs, these services allow for third party developers to build upon these services as well as extend them. This paper evaluates an application built to utilize multiple location-based service APIs in order to provide an all-in-one check-in solution for an Android-based mobile phone.

# BACKGROUND

This is the background section

To date, numerous Location-based mobile applications exist for the Android and iOS platforms. These applications include Foursquare, Gowalla, Brightkite, Yelp, and SCVNGR, just to name a few. These applications serve many different purposes. For instance, Yelp is primarily a review site for businesses. Where as SCVNGR is geared towards scavenger hunts, location challenges, and adventuring. Foursquare, Gowalla, and Brightkite lean more towards the social networking side of applications. With these three services, users maintain friendships and interact with their friends through messaging or multimedia. Although these products serve different primary purposes, they also share a lot of location functionality in common. Many provide maps of where users are located and what is around them. Another feature in common is the gaming aspect of location-based applications. Many reward users with virtual prizes for going to certain places or doing certain things. Another thing they all share in common, and what seems to be the most popular feature of the moment, is the users’ ability to “check-in” at their current location. Though it may be called something different across applications, the idea of “checking in” is the same, and it gives the user the ability to mark their current location at the current time. They can then notify their friends of their location or allow others to see their current location. To some mobile phone users, this seems like the ultimate invasion of privacy; the ability of anyone to track your personal movements throughout the day does not appeal to everyone. However, to many, the ability to “check-in” has many useful benefits. It allows users to let their friends know that they are currently at a bar or night club so others can come join them. This can take a lot of the phone calls or text messages out of planning and organizing a night out [3]. Checking in at certain locations can provide useful information to the user as well. If a user is on vacation and “checks in” at a nearby restaurant, patrons that have previously been to the restaurant can provide information on the best dish or what food to avoid. The vacationer can even use it to decide where the best meal is located before they even “check-in”. Advertising can also be made useful, rather than annoying, via location-based information. It has been shown that users actually respond well to location-based ads [1], similar to the way targeted ads on the web work better than random ones. Foursquare, for example, has a built-in specials capability for businesses with locations on Foursquare. A manager can offer discounts to the patron with the most “check-ins” at the business. Finally, the “check-in” ability is just fun, which is what appeals to many users. It let’s the user keep track of where they have been, earn prizes, and keep in touch and play games with friends.

However, as useful and fun as “checking in” is on all of these location-based applications, there is still a major problem; there are too many location-based applications to “check-in” with. Unlike websites such as Google, the king of search, Facebook, the king of social interaction, or Wikipedia, the king of information, there has yet to be an application crowned king of the location-based mobile applications. Foursquare is arguably the most popular, however, since this is a fast growing and very popular market, many big names are jumping into area as well. Facebook, for instance, just launched its “Places” functionality over the summer, allowing users to “check-in” while posting their statuses from their iPhones. With so many popular new players and established big names in the mix, there are currently too many applications to “check-in” with and no way of deciding which is going to be the most popular in the future. This essentially forces users to choose between a select few of the location services since nobody wants to take out their phone and “check-in” on 10+ different applications every time they go to a new location. It would be far too time consuming. Unfortunately, it is still difficult to determine which service will ultimately win the popularity race, and many people have friends spread across multiple services. CheckIn4Me aims to provide a solution to the problem of “checking in” over multiple applications and services. CheckIn4Me will allow users to connect their location-based service accounts to CheckIn4Me, via the services’ own APIs, and then “check-in” once from within CheckIn4Me across all of their connected services. This provides a quick way to “check-in” while not limiting the services with which the users can share their information.

# SYSTEM

CheckIn4Me utilizes a number of technologies to achieve its goal as an application. First and foremost, CheckIn4Me runs on the Android mobile operating system. Thus, it uses the Android software development kit provided by Google. Android is a Linux-based operating system. The Android software development kit provides all the tools a user would need to build an Android application. It includes libraries, a debugger, and an emulator, all of which integrate into the Eclipse IDE quite well. All applications, including CheckIn4Me, are written in Java and run on the Dalvik virtual machine.

For connecting services to the CheckIn4Me, the OAuth protocol was used. OAuth was chosen because most of the APIs provided by location-based services implement some version of it for connecting to their services. OAuth allows users of CheckIn4Me to authorize the use of API calls on the user’s behalf without exchanging any username or password information with CheckIn4Me itself. Multiple versions of OAuth are used in CheckIn4Me. For example, Foursquare implements OAuth version 1.0, whereas Gowalla implements a draft version of the upcoming OAuth version 2.0. Though very different in the details, they generally work as follows: The application requests a token from the service implementing OAuth. The application then sends the user to the service’s authorization website for OAuth with the token and a redirect callback. When the user authorizes the application to make API calls on the user’s behalf, the service redirects the user back to the application using the callback and returns the token. The application then exchanges the token for an access token. Once the application has this token, it provides it in all API calls to the service in order to retrieve and publish user data.

For interacting with multiple location-based services, CheckIn4Me contains implementations of each service’s public API. Using the public API with OAuth protocol allows the application to retrieve locations from the services based on the Longitude and Latitude coordinates provided by the Android operating system (which will gather data from a GPS provider or a network provider) and allows the application to check-in for a user by sending the service the user’s desired check-in location.

sequence diagram from websequencediagrams.com

uml diagram on http://www.gliffy.com/

Discussion / Future Work / Limitations

Future work for CheckIn4Me includes implementing some of the major features existing on the services that are still missing from CheckIn4Me. These features include posting a message when checking in, posting a photo from a check-in location, as well as searching for locations in the nearby area. Future work also includes incorporating other services. For the purpose of this project and paper, CheckIn4Me only implemented Foursquare and Gowalla API functionality. The idea is to incorporate as many location-based services as possible, which includes Facebook Places, Brightkite, Yelp, SCVNGR, etc.

This is the system section – 2 figures needed

- screen shot?

- flow control?

- Android SDK / JAVA / Eclipse

- class diagram

- data normalization

- HTTP

- Foursquare API

- Gowalla API

- JSON data exchange

- OAuth authentication

# EVALUATION

This is the evaluation section

- how best be tested with real users

- test results on 4 to 6 users

- interview gf, mom, dad, 2 friends, coworker

- hard figures

- user comments

- performance of oauth

- relying on external browser

- api rate limits

- performance of multiple http calls in succession

# DISCUSSION

This is the discussion section

- apis provide staying power for services

- sdks and apis provide ease of development

- freedom of service choice

- freedom of sharing choice

# REFERENCES

1. Rashid, O., Coulton, P., Edwards, R. Providing Location Based Information/Advertising for Existing Mobile Phone Users. *Springer-Verlag London Limited.* (November 2006) 3-10.
2. Rashid, O., Mullins, I., Coulton, P., Edwards, R. Extending Cyber Space: Location Based Games Using Cellular Phones. *ACM Computers in Entertainment 4, 1* (January 2006) 1-18.
3. Weilenmann, A., Leuchovius, P. “I’m waiting where we met last time”: Exploring Everyday Positioning Practices to Inform Design. *NordiCHI ’04.* (October 2004) 33-42.
4. Schmidt, A., Holleis, P., Hakkila, J., Rukzio, E., Atterer, R. Mobile Phones as Tool to Increase Communication and Location Awareness of Users. *Mobility 06* (October 2006) 1-7.
5. Bently, F., Metcalf, C. Location and Activity Sharing in Everyday Mobile Communication. *CHI 2008.* (April 2008) 2453-2462.
6. Sohn, T., Griswold, W., Scott, J., LaMarca, A., Chawathe, Y., Smith, I., Chen, M. Experiences with Police Lab: An Open Source Toolkit for Location-Aware Computing. *ICSE’06* (May 2006) 462-471.
7. Misund, G., Holone, H., Karlsen, J., Tolsby, H. Chase and Catch – Simple as That? Old-fashioned Fun With Location-Aware Mobile Phones. *ACM* (October 2009) 73-80.
8. Seeburger, J., Schroeter, R. Disposable Maps: Ad hoc Location Sharing. *ACM.* (2009) 377-380.
9. Dearman, D., Hawkey, K., Inkpen, K. Effect of Location-Awareness on Rendezvous Behaviour. *CHI 2005* (April 2005) 1929-1932.
10. O’Hara, K., Kindberg, T., Glancy, M., Baptista, L., Sukumaran, B., Kahana, G., Rowbotham, J. Social Practices in Location-Based Collecting. *CHI 2007.* (April – May 2007) 1225-1234.
11. Jedrzejczyk, L., Price, B., Bandara, A., Nuseibeh, B. On the Impact of Real-Time Feedback on Users’ Behaviour in Mobile Location-Sharing Applications. *Symposium On Usable Privacy and Security 2010.* (July 2010).
12. Church, K., Neumann, J., Cherubini, M., Oliver, N. The “Map Trap”? An Evaluation of Map Versus Text-Based Interfaces for Location-Based Mobile Search Services. *International World Wide Web Conference Committee.* (April 2010) 261-270.
13. Chow, C., Mokbel, M. Privacy in Location-Based Services: A System Architecture Perspective. 23-27.
14. Cai, Y., Xu, T. Design, Analysis, and Implementation of a Large-Scale Real-Time Location-Based Information Sharing System. *MobiSys ’08.* (June 2008) 106-117.
15. Lemmela, S., Korhonen, H. Finding Communication Hot Spots of Location-Based Posting. *CHI 2007.* (April – May, 2007) 2549-2554.

Notes to self – pdf file names:

779

p3c

a21

p33

p2453

p462

p73

p377

p1929

p1225

a14

p261

p23

p106

p2549